



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

INFORMATION TECHNOLOGY P1

NOVEMBER 2014

MARKS: 150

TIME: 3 hours

This question paper consists of 22 pages.

INSTRUCTIONS AND INFORMATION

1. This paper is divided into THREE sections. Candidates must answer ALL THREE 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 paper is set in programming terms that are not specific to any particular programming language (Delphi/Java (using the Netbeans IDE)).
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 only.
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. You may not use components provided within the interface to store and later retrieve data.
9. You must save your work regularly on the disk you have been given, or 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 disc 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.

13. The files you need to complete this question paper have been given to you on a disk/CD/DVD/flash disk or the disk space allocated to you in the form of a password-protected executable file:

- Delphi learners must use the file **DelphiDataENG.exe**
- Java learners must use the file **JavaDataENG.exe**

Do the following:

- Double click on the file
- Click on the extract button
- Enter the following password: **Transport@(!\$**

List of files provided in the folder DelphiDataENG/JavaDataENG (once extracted):

Delphi files

Question1:

Question1_P.dpr
Question1_P.res
Question1_U.dfm
Question1_U.pas

Question2:

Delivery_U.pas
DeliveryInfo.txt
Question2_P.dpr
Question2_P.res
Question2_U.dfm
Question2_U.pas

Question3:

Question3_P.dpr
Question3_P.res
Question3_U.dfm
Question3_U.pas

Java (Netbeans) files

Question1:

Question1.form
Question1.java

Question2:

Delivery.java
DeliveryInfo.txt
Question2.form
Question2.java

Question3:

Question3.form
Question3.java

SCENARIO:

SuperTrans Courier Services is a national transport company with branches throughout South Africa. You are requested to assist with some of the software applications the company intends to implement shortly.

SECTION A**QUESTION 1: GENERAL PROGRAMMING SKILLS****INSTRUCTIONS:**

Delphi programmers	Java programmers
<ul style="list-style-type: none"> The project Question1 is provided in the DelphiDataENG folder. Open the incomplete project file Question1_P.dpr in the Question1 folder. Add your examination number as a comment in the first line of the main form unit (Question1_U) file. 	<ul style="list-style-type: none"> The project Question1 is provided in the JavaDataENG folder. Open the incomplete class called Question1.java contained in the folders Source Packages (src), Question1Package. Add your examination number as a comment in the first line of the class (Question1).

Do the following:

- Compile and execute the program. The interface displays five different sections labelled Question 1.1 to Question 1.5. The program currently has no functionality. An example of the interface is given below:

The screenshot shows a software interface with five distinct sections:

- Question 1.1:** Contains two dropdown menus for 'Select the place of departure' and 'Select the destination', both set to 'Cape Town'. Below them is a text input field for 'Enter the number of kilometres'. At the bottom left is a 'Confirm delivery' button, and at the bottom right is a 'Delivery label' button.
- Question 1.2:** Features a list box for 'Select the category' with options A1 (0kg - 1kg), A2 (>1kg - 2kg), A3 (>2kg - 5kg), and A4 (>5kg). To the right is a checkbox for 'Tick if speed post is required' with the label 'Speed post'. Below is a 'Delivery cost' button and a text input field.
- Question 1.3:** Contains a 'Delivery box number' button and a text input field.
- Question 1.4:** Has a 'UPC bar code' text input field containing '639382000393' and a 'Validate bar code' button. Below is a large empty text area.
- Question 1.5:** Includes a 'Select a city' dropdown menu set to 'Cape Town' and a 'View and save deliveries' button. Below is a large empty text area.

- Complete the code for each section of QUESTION 1 as described in QUESTIONS 1.1 to QUESTION 1.5 below.

1.1 **Button – [Confirm Delivery]**

Obtain the following data from the relevant components:

- Place of departure from the **Place of Departure** combo box
- Destination from the **Destination** combo box
- Number of kilometres from the **Kilometres** text box

Create a line of text as output that indicates the place of departure, the destination and the number of kilometres as shown in the example below. Place the constructed line of text in the label component provided.

Example of possible input:

Select the place of departure	Johannesburg
Select the destination	Durban
Enter the number of kilometres	635

Required output:

Delivery label
Johannesburg to Durban : 635 km

(5)

1.2 **Button – [Delivery cost]**

The following components are provided:

- A list box that indicates the weight categories of deliveries in terms of codes (A1–A4) in the following format:

<code><space><(range in kg)>

A1 (0kg - 1kg)
A2 (>1kg - 2kg)
A3 (>2kg - 5kg)
A4 (>5kg)

The following tariffs per weight category apply:

Category	Tariff per kilometre
A1	R0,60
A2	R1,00
A3	R1,25
A4	R1,65

- A check box that indicates whether speed post must be used. A standard amount of R100,00 is charged for speed post.

When the user clicks on the **Delivery cost** button the number of kilometres entered in QUESTION 1.1, the selected weight category and whether speed post is required or not must be used to calculate the delivery cost.

Example of the output if the number of kilometres is 635, the weight of the item to be delivered is in the A2 category and speed post is required:

The screenshot shows a form titled "Question 1.2". It contains two main sections. The first section is "Select the category" with a dropdown menu listing four options: "A1 (0kg - 1kg)", "A2 (>1kg - 2kg)", "A3 (>2kg - 5kg)", and "A4 (>5kg)". The "A2 (>1kg - 2kg)" option is selected. The second section is "Tick if speed post is required" with a checked checkbox labeled "Speed post". Below these sections is a "Delivery cost" button and a text box displaying "R735.00".

(10)

1.3 Button – [Delivery box number]

Items to be delivered need to be placed in specific delivery boxes. The correct delivery box for each individual item must be determined using the criteria below:

- There are five delivery boxes numbered from 1 to 5.
- All **speed post items** will be placed in delivery box **4**.
- All the other delivery items will be **randomly** placed in the remaining delivery boxes (1, 2, 3 or 5).

Display the number of the delivery box into which the item must be placed.

Example of output if speed post was requested in QUESTION 1.2 (on the next page):

Question 1.2

Select the category

- A1 (0kg - 1kg)
- A2 (>1kg - 2kg)
- A3 (>2kg - 5kg)
- A4 (>5kg)

Tick if speed post is required

Speed post

Delivery cost: R735.00

Question 1.3

Delivery box number: 4

Example of output when speed delivery is NOT required in QUESTION 1.2:

Question 1.2

Select the category

- A1 (0kg - 1kg)
- A2 (>1kg - 2kg)
- A3 (>2kg - 5kg)
- A4 (>5kg)

Tick if speed post is required

Speed post

Delivery cost: R635.00

Question 1.3

Delivery box number: 3

NOTE: Due to the nature of the random function the value for the delivery box displayed in the screenshot above may differ from the value displayed by your program.

(9)

1.4 **Button – [Validate bar code]**

A Universal Product Code (UPC) bar code is printed on items to be delivered. The picture on the next page shows an example of a UPC bar code:



The bar code number consists of twelve digits. For example, the bar code number shown in the picture above is 639382000393. The last digit of a UPC is called a check digit. The check digit is used by a scanner to determine whether a bar code is valid or not.

Use the algorithm below to write code for the **Validate bar code** button. The code needs to verify the check digit and display a message to indicate whether the bar code is valid or not.

Algorithm:

1. barCode \leftarrow text box value
2. sumOddPositions \leftarrow 0
3. sumEvenPositions \leftarrow 0
4. loop from first digit to second last digit of barCode
5. if the logical position of the digit in the bar code is even
 sumEvenPositions \leftarrow sumEvenPositions + digit at position
 else
 sumOddPositions \leftarrow sumOddPositions + digit at position
6. sum \leftarrow sumOddPositions * 3 + sumEvenPositions
7. checkDigit \leftarrow 10 – (sum modulus 10)
8. if checkDigit = last digit of barCode
 display the check digit and a suitable message indicating that the bar code is valid
 else
 display a suitable message indicating that the bar code is invalid

Example:

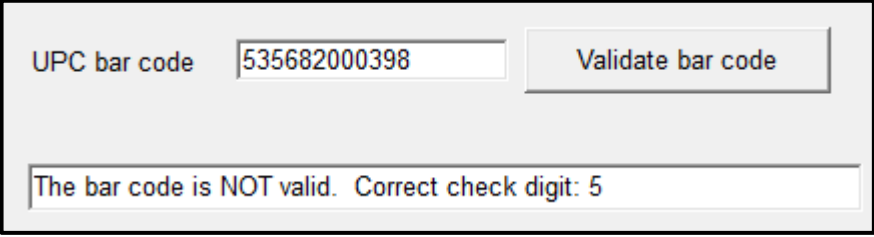
If the bar code is 639382000393 then:

$$\text{sumOddPositions} = 6+9+8+0+0+9 = 32$$

$$\text{sumEvenPositions} = 3+3+2+0+3 = 11$$

Example of a possible output for a valid bar code:

Example of a possible output for an invalid bar code:



UPC bar code 535682000398 Validate bar code

The bar code is NOT valid. Correct check digit: 5

(14)

1.5 Button – [View and save deliveries]

All deliveries for December 2014 are stored in the given array called **arrDecDeliveries**. The format of each entry in the array is as follows:

<date><space><place of departure><space>to<space><destination>

Example:

2014-12-01 Durban to Cape Town

The user has to select a city from the provided combo box. All the deliveries that were made during December 2014 to or from the selected city must be displayed in the provided output area and also written to a text file.

Write code to do the following:

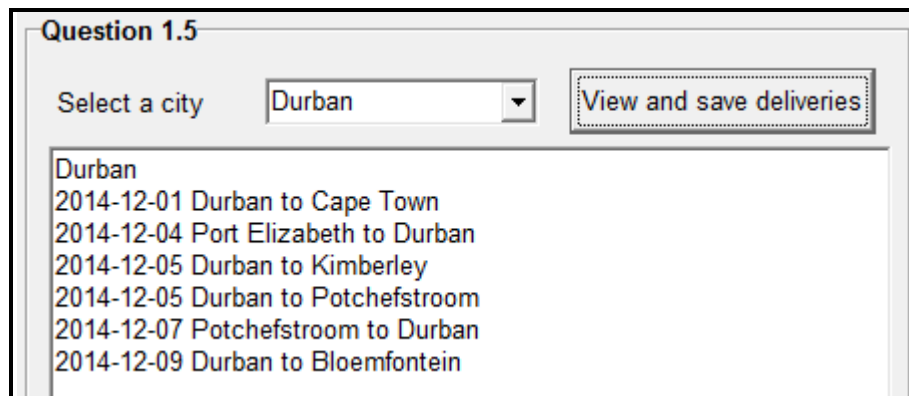
- Create a text file where the name of the file is made up of the text 'December2014' combined with the name of the city that was selected.

Example:

If Durban was selected, the name of the text file must be **December2014Durban.txt**.

- Use the data stored in the **arrDecDeliveries** array and display the deliveries to and from the selected city in the output area provided. Use the name of the city as a heading.
- Store the deliveries to and from the selected city in the text file that was created – one delivery per line.

Example of the contents of the output area if the city of Durban is selected:



Question 1.5

Select a city

Durban
2014-12-01 Durban to Cape Town
2014-12-04 Port Elizabeth to Durban
2014-12-05 Durban to Kimberley
2014-12-05 Durban to Potchefstroom
2014-12-07 Potchefstroom to Durban
2014-12-09 Durban to Bloemfontein

Example of the contents of the **December2014Durban.txt** text file:

```
2014-12-01 Durban to Cape Town
2014-12-04 Port Elizabeth to Durban
2014-12-05 Durban to Kimberley
2014-12-05 Durban to Potchefstroom
2014-12-07 Potchefstroom to Durban
2014-12-09 Durban to Bloemfontein
```

(12)

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

TOTAL SECTION A: 50

SECTION B**QUESTION 2: OBJECT-ORIENTATED PROGRAMMING**

SuperTrans Courier Services owns five trucks. The trucks are classified as light, medium or heavy-duty trucks. The trucks are used on four different routes (RN1 to RN4) to deliver cargo.

INSTRUCTIONS:

Delphi programmers	Java programmers
<p>The project Question2 is provided in the DelphiDataENG folder which contains:</p> <ul style="list-style-type: none"> ○ A main form unit called Question2_U.pas ○ An incomplete unit file called Delivery_U.pas ○ A text file (DeliveryInfo.txt) that contains information on deliveries <ul style="list-style-type: none"> • Open the incomplete project file called Question2_P.dpr in the Question2 folder. • View (Ctrl + F12) the unit file Delivery_U.pas and add your examination number as a comment in the first line of both files Question2_U.pas and Delivery_U.pas. 	<p>The project Question2 is provided to you in the JavaDataENG folder which contains:</p> <ul style="list-style-type: none"> ○ A GUI class file called Question2.java ○ An incomplete object class file called Delivery.java ○ A text file (DeliveryInfo.txt) that contains information on deliveries <ul style="list-style-type: none"> • Open the incomplete classes called Question2.java and Delivery.java in the folders Source Packages (src), Question2Package. • Add your examination number as a comment in the first line of both classes Question2.java and Delivery.java.

Do the following:

- Compile and execute the program. The program currently has no functionality. An example of the interface is shown on the next page:

- Complete the code for this program as specified in QUESTION 2.1 and QUESTION 2.2 below.

2.1 The given incomplete object class (**TDelivery/Delivery**) contains the following code:

- The declaration of five attributes which describes a **delivery** object
- The declaration of a two-dimensional array to be used to determine toll fees
- A **toString** method

The attributes of a **delivery** object are the following:

Names of attributes		Description
Delphi	Java	
fDeliveryNum	deliveryNum	Number assigned to a specific delivery, for example 1, 2, et cetera
fTruckNum	truckNum	Truck number, for example Tr1, Tr2, Tr3, Tr4 or Tr5
fFuelUsed	fuelUsed	Fuel used during the delivery
fOdoStart	odoStart	Odometer reading at the start of the delivery
fOdoEnd	odoEnd	Odometer reading on completion of the delivery

Complete the code in the given **delivery** class (**TDelivery/Delivery**) as described in QUESTION 2.1.1 to QUESTION 2.1.4 below:

2.1.1 Write code for a constructor method to receive the delivery number, truck number, odometer reading at the start of the delivery and odometer reading on completion of the delivery as parameter values. Assign these values to the relevant attributes of the object class. (3)

2.1.2 Write a mutator method and an accessor method for the **fFuelUsed/fuelUsed** attribute. (4)

2.1.3 Write a method called **calculateDistance** to calculate and return the distance travelled based on the start and end odometer readings for the delivery. (3)

2.1.4 Different toll fees must be paid on different toll routes. The routes used are RN1, RN2, RN3 and RN4. Toll fees on these routes are dependent on the type of truck used. The company's trucks are classified as follows:

- Light-duty trucks: Tr1, Tr2
- Medium-duty truck: Tr3
- Heavy-duty trucks: Tr4, Tr5

A two-dimensional array called **tollFees** contains the toll fees for the different routes for different types of trucks and is supplied as part of the given code. The contents of the array can be represented as follows:

	Light-duty truck	Medium-duty truck	Heavy-duty truck
RN1	R105,50	R135,00	R210,00
RN2	R35,00	R54,00	R82,00
RN3	R85,00	R129,00	R205,00
RN4	R112,00	R170,00	R219,00

Rows: Represents the routes RN1 to RN4

Columns: Represents the types of trucks

Write a method called **determineTollFees** to determine and return the toll fees to be paid for the delivery. The method must receive the route (RN1, RN2, RN3 or RN4) as a parameter. Use the two-dimensional array called **tollFees** to look up the toll fee for the route and type of truck that was used for the delivery.

NOTE: It is compulsory to use the given two-dimensional array in your solution to look up the toll fee. (10)

- 2.2 A text file named **DeliveryInfo.txt** contains an unknown number of lines of information on previously completed deliveries. Each line of information contains data on a single delivery in the following format:

<delivery number>#<truck number>#<odometer reading on completion of the delivery>

Example of some of the data in the text file named **DeliveryInfo.txt**:

```
1#Tr1#121110
2#Tr2#8010
3#Tr3#15021
4#Tr4#700
5#Tr1#121453
6#Tr3#15653
:
```

The data of the first delivery can be interpreted as follows:

- Delivery **1** identifies the delivery.
- Truck **Tr1** was used to make the delivery.
- The reading on the odometer on completion of the delivery was **121110**.

Do the following to complete the code for each button in the main form unit (Delphi)/GUI class (Java) as described below:

2.2.1 Button – [Get data from file]

- Select the specific truck to be used from the provided **Truck number** combo box.
- Use the given text file **DeliveryInfo.txt** to determine the following:
 - **New delivery number**
The delivery number for the new delivery will follow on the number of the last delivery stored in the text file. If the information of 20 deliveries are stored in the text file, the number of the new delivery will be 21.
 - **Start odometer reading**
The odometer reading for the last delivery that was made by the selected truck must be used as the start odometer reading for the new delivery.

Example:

If truck Tr4 is selected and the odometer reading that was captured in the text file on completion of the last delivery for Tr4 is 1648, then the value of 1648 must be used as the odometer reading at the start of the new delivery.

- Display the **new delivery number** and **start odometer reading** in the text boxes provided.
- Display a suitable message in a dialog box and terminate the program if the text file cannot be accessed.

Example of the content of the relevant text boxes when truck Tr4 is selected:

The screenshot shows a dialog box with the following elements:

- Title: **Create and display new delivery object**
- Dropdown menu: **Select truck number** (Tr4)
- Button: **Get data from file**
- Text input: **New delivery number** (21)
- Text input: **Start odometer reading** (1648)
- Text input: **Enter end odometer reading** (empty)
- Buttons: **New delivery** and **Display delivery**

NOTE: If you are unable to read the required information from the given text file, type in the number and the start odometer reading for the delivery into the text boxes to continue with the rest of the program.

(14)

2.2.2 Button – [New delivery]

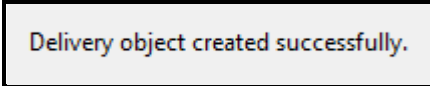
To instantiate a new **Delivery** object, the user needs to first enter the end odometer reading for the delivery in the textbox provided.

NOTE: The end odometer reading entered by the user must be greater than the previous odometer reading of the truck, because it is the start odometer reading for this delivery. No validation is necessary.

NOTE: The **Delivery** object variable has already been declared globally as part of the given code.

Write code for the **New delivery** button to do the following:

- Use the new delivery number, the truck number, the odometer reading at the start and the odometer reading at the end of the delivery to instantiate the **Delivery** object.
- Display a message indicating that the object has been instantiated successfully.



Delivery object created successfully.

- Obtain the distance travelled by calling the **calculateDistance** method and calculate the estimated litres of fuel used for the delivery. Use the following information:

One litre of fuel is used for every five kilometres travelled.

Use the method you have written in QUESTION 2.1 to set the **fFuelUsed/fuelUsed** attribute to this calculated value.

NOTE: The following buttons must be enabled:

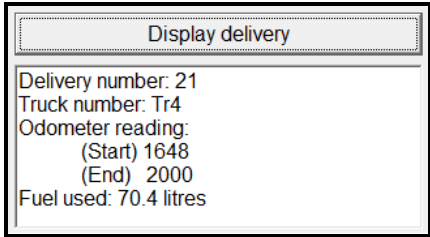
- Check fuel used
- Calculate toll fees

(10)

2.2.3 Button – [Display delivery]

Display the object in the output area provided using the **toString** method.

Example of the output for truck Tr4 with an end odometer reading of 2000:



Display delivery

Delivery number: 21
Truck number: Tr4
Odometer reading:
 (Start) 1648
 (End) 2000
Fuel used: 70.4 litres

(2)

2.2.4 Button – [Check fuel used]

The driver of the truck is required to refill the fuel tank at the end of each delivery. Due to the nature of the routes, the amount of fuel used to refill the tank may sometimes differ from the fuel calculated for the delivery.

Write code to do the following:

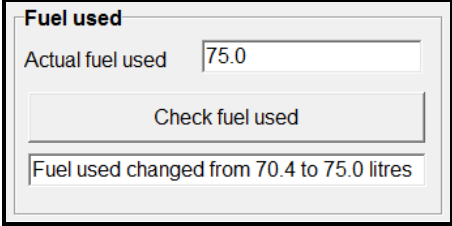
- Enter the amount of fuel used to refill the tank in the text box provided.

HINT: The value of 75 can be used to test the code.

- Obtain the calculated amount of fuel used from the relevant attribute in the object class.
- Calculate the difference between the fuel used to refill the tank and the calculated fuel used as stored in the attribute of the **Delivery** object.
 - If the difference is less than 10%, the **fFuelUsed/fuelUsed** attribute must be updated by setting its value to the new value as entered in the text box.

An appropriate message must be displayed in the text box provided, indicating whether the value of the **fFuelUsed/fuelUsed** attribute has been updated or not.

Example of the output if the delivery was done by truck Tr4 and the actual fuel used is 75,0:



The screenshot shows a window titled "Fuel used". Inside the window, there is a label "Actual fuel used" followed by a text input field containing the value "75.0". Below the input field is a button labeled "Check fuel used". At the bottom of the window, there is a message box that says "Fuel used changed from 70.4 to 75.0 litres".

- If the difference is greater than or equal to 10%, an error message should be displayed and the value of the **fFuelUsed/fuelUsed** attribute must not be updated.

Example of the output if the delivery was done by truck Tr4 and the actual fuel used is 85,0:

(9)

2.2.5 Button – [Calculate toll fees]

To determine the toll fees for the route used for the new delivery, the route number (for example RN3) must be entered in the text box provided. This route number must be sent as a parameter to the **determineTollFees** method.

The toll fees must be displayed in the label component provided. The amount must be displayed as currency (rand) to TWO decimal places.

Example of the output if the delivery was done by truck Tr4 on route RN3:

(5)

- Ensure that your examination number is entered as a comment in the first line of the class as well as the form.
- Save all the files.
- Print code you created for both the classes, if printouts are required.

TOTAL SECTION B: 60

SECTION C

QUESTION 3: PROBLEM-SOLVING

SuperTrans Courier Services provides a daily speed service from Cape Town to Johannesburg. The storage space in the truck used for the deliveries is divided into two shelves for fragile and non-fragile items respectively. The maximum capacity of the shelf reserved for fragile items is 20 items while a maximum of 30 items can be placed on the shelf reserved for non-fragile items. The loading zone at the company controls the clearing of a load, loading of items into the truck and checking the status of the truck load at any given time.

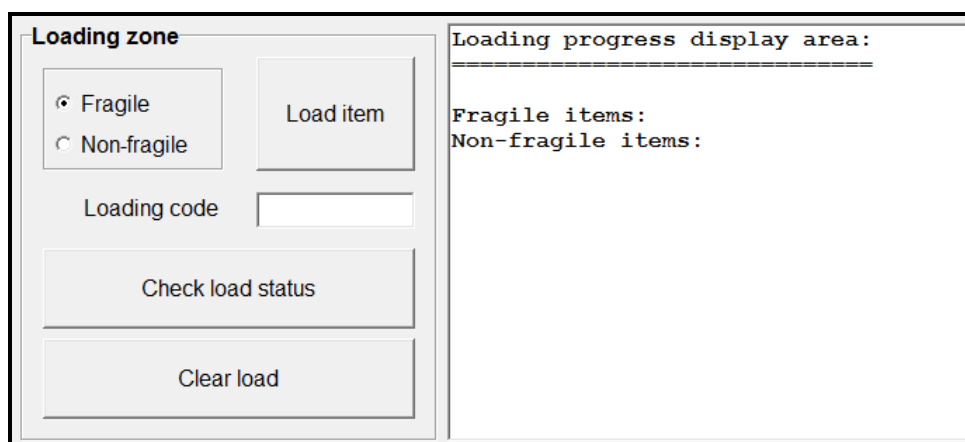
An incomplete program has been provided to manage the loading of trucks.

INSTRUCTIONS:

Delphi programmers	Java programmers
<ul style="list-style-type: none"> The project Question3 is provided in the DelphiDataENG folder. Open the incomplete project file Question3_P.dpr in the Question3 folder. Add your examination number as a comment in the first line of the main form unit (Question3_U) files. 	<ul style="list-style-type: none"> The project Question3 is provided in the JavaDataENG folder. Open the incomplete class called Question3.java in the folders Source Packages (src), Question3Package. Add your examination number as a comment in the first line of the class (Question3).

Do the following:

- Compile and execute the program. The program currently has no functionality. An example of the interface is given below.



The output area of the GUI labelled 'Loading progress display area' represents the storage area of the truck. An asterisk (*) is used to represent an item placed on a shelf.

Example of the display of a fully loaded truck with 20 fragile and 30 non-fragile items:

```

Loading progress display area:
=====
Fragile items:      *****
Non-fragile items: *****

```

- Write code to complete QUESTION 3.1 to QUESTION 3.3 as explained in the instructions below:

3.1 **Button – [Load item]**

Write code to do the following when an item is loaded:

- Create a loading code.
- Add the item, represented by an asterisk (*), to the correct shelf in the loading progress display area.

Loading code:

The loading code is compiled using the letter 'F' for fragile items and the letters 'NF' for non-fragile items, followed by the sequence number of the item on the shelf.

For example:

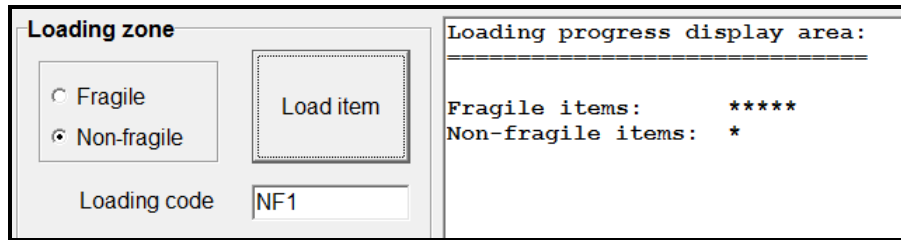
F1 refers to item 1 on the shelf for fragile items.
NF6 refers to item 6 on the shelf for non-fragile items.

Add item to loading progress display area:

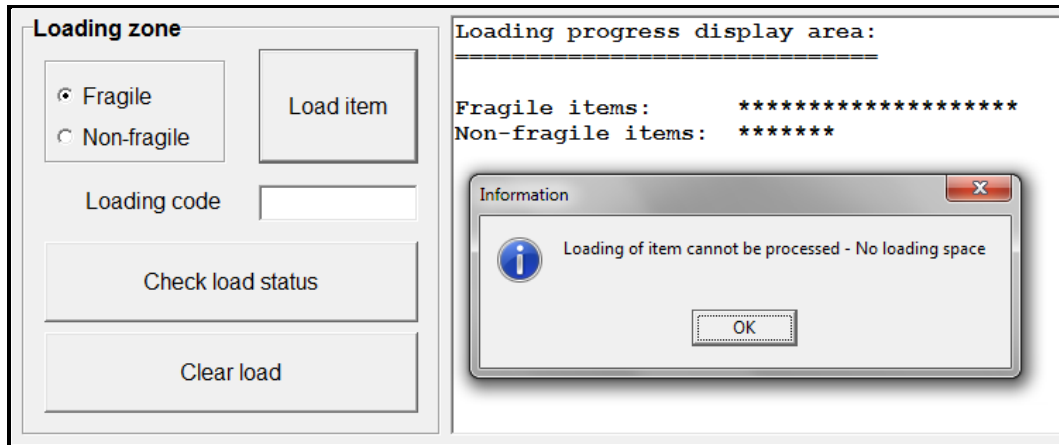
If there is space on the shelf, create and display the loading code in the text box provided and update the loading progress display area to show the new item.

If the item cannot be loaded, the loading code must be left empty and a dialog box must be used to display the following message: '**Loading of item cannot be processed – No loading space**'

Example of output of the 'Loading progress display area' if the item loaded is the first non-fragile item. In this example five fragile items have already been loaded (on the next page):



Example of an attempt to load a fragile item when the shelf for fragile items is full:



(20)

3.2 **NOTE:** If you could not complete QUESTION 3.1, use the data below to complete QUESTION 3.2:

Number of fragile items: 4
Number of non-fragile items: 13

Button – [Check load status]

The delivery can be made if the truck has a minimum load of:

- 50% fragile items (10 fragile items), and
- 50% non-fragile items (15 non-fragile items)

When this button is clicked, a load status report must display a summary of the number of both fragile and non-fragile items, using the following column headings:

Item type	Number of items	Percentage loaded
-----------	-----------------	-------------------

- If the percentage of items loaded is 50% or more for both fragile and non-fragile items, the message **'The delivery may progress'** must be displayed.

- If the percentage of items loaded is less than 50% for fragile OR non-fragile items, then the message '**The delivery may not progress**' must be displayed along with the outstanding number of items for each type required to make up the minimum load.

NOTE: Marks will be awarded for column formatting and for displaying the percentage loaded to TWO decimal places.

Example 1: If the load status of the truck is 4 fragile items and 10 non-fragile items, the load status report should be as follows:

```
Load status report:
=====
Item type      Number of items      Percentage loaded
Fragile        4                    20.00
Non-fragile    10                   33.33

The delivery may not progress.
Number of fragile items still required: 6
Number of non-fragile items still required: 5
```

Example 2: If the load status of the truck is 12 fragile items and 17 non-fragile items, the load status report should be as follows:

```
Load status report:
=====
Item type      Number of items      Percentage loaded
Fragile        12                   60.00
Non-fragile    17                   56.67

The delivery may progress.
```

(17)

3.3 Button – [Clear load]

Initialise all variables and data structures to prepare for a new load. Also clear all text from the output area.

(3)

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

TOTAL SECTION C: 40
GRAND TOTAL: 150



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

INFORMATION TECHNOLOGY P1

NOVEMBER 2014

MEMORANDUM

MARKS: 150

This memorandum consists of 28 pages.

GENERAL INFORMATION:

- These marking guidelines are to be used as the basis for the marking session. They were prepared for use by markers. All markers are required to attend a rigorous standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' work..
- Note that learners who provide an alternate correct solution to that given as example of a solution in the marking guidelines will be given full credit for the relevant solution, unless the specific instructions in the paper was not followed or the requirements of the question was not met
- **Annexures A, B and C** (pages 3-8) include the marking grid for each question for using either one of the two programming languages.
- **Annexures D, E, and F** (pages 9-16) contain examples of solutions for Java for Questions 1 to 3 in programming code.
- **Annexures G, H and I** (pages 17-28) contain examples of solutions for Delphi for Questions 1 to 3 in programming code.
- Copies of **Annexures A, B and C** (pages 3-8) should be made for each learner and completed during the marking session.

ANNEXURE A:**SECTION A:****QUESTION 1: MARKING GRID - GENERAL PROGRAMMING SKILLS**

CENTRE NUMBER:		EXAMINATION NUMBER:	
QUESTION	DESCRIPTION	MAX. MARKS	LEARNER'S MARKS
	<i>If a learner has a problem reading from a combo box, penalise only once for the error.</i>		
1.1	Button - [Confirm delivery] Extract departure from combobox; (to be used as text) AND Extract destination from combobox; (to be used as text)✓ Extract the number of kilometres from text box✓ convert to number✓ (can also be converted in 1.2) Create the string to join departure, destination and distance✓ and assign to the label provided✓	5	
1.2	Button - [Delivery cost] Extract the choice selected from the list box✓ Correct variable in condition✓ All 4 possibilities (A1..A4 OR item index 0..3)✓ Correct selection structure (if/case/switch)✓ Use the correct tariff for each option✓ Calculate cost: tariff x distance✓ Check if speed post is selected✓, add 100 to cost✓ Set the cost to the text box provided✓ formatted to 1 or 2 decimal places✓ (Accept solutions that did not consider the leading spaces)	10	
1.3	Button - [Delivery box number] Create variable to store box number ✓ Check if speed post is selected✓ Set box number = 4 ✓ else✓ (could be another if statement) Generate random number ✓ in the correct range✓ Ensure that the value is not 4 ✓ ✓ (Only one mark if generated once more) Display box number✓	9	

QUESTION 1: MARKING GRID - GENERAL PROGRAMMING SKILLS (continue)

1.4	<p>Button - [Validate bar code] Extract the barcode ✓ Set sumOdd and sumEven to 0 ✓ Loop ✓ from first to the second last digit ✓ Check if the position of the digit is even ✓ Add value of the digit ✓ (integer) at position to sumEven ✓ Else Add value of the digit at position to sumOdd ✓ Multiply sumOdd by 3 ✓ Add sumEven and sumOdd ✓ Calculate checkdigit: Subtract total modulus 10 ✓ from 10 ✓ If checkdigit equals last digit ✓ (Must be same data types) Display appropriate message that bar code is valid } ✓ Else Display message that the bar code is not valid } ✓ (Display of the check digit not necessary for the if or else)</p>	14	
1.5	<p>Button - [View and save deliveries] Extract the selected name of the city from the combo box ✓ Display the name of the city in the output area as a heading ✓ Create a text file ✓ with the correctly constructed name ✓ Loop from first position ✓ to last position in the array ✓ Check if city name is part ✓ of the correct array entry ✓ Display the entry in the output area if found ✓ and store the entry in the text file ✓ One delivery per line ✓ Close the text file outside loop ✓</p>	12	
	TOTAL:	50	

ANNEXURE B:**SECTION B:****QUESTION 2: MARKING GRID - OBJECT-ORIENTED PROGRAMMING**

CENTRE NUMBER:		EXAMINATION NUMBER:	
QUESTION	DESCRIPTION	MAX. MARKS	LEARNER'S MARKS
2.1.1	Constructor: Heading with ONLY four values✓ Correct data types✓ Assign parameter values to four attributes✓ (default for fuelUsed can be included)	3	
2.1.2	Accessor and mutator METHODS: setFuelUsed (non return)✓ with parameter value assigned to attribute✓ getFuelUsed(return)✓ with correct return data type✓	4	
2.1.3	calculateDistance METHOD: return type a number✓ subtract odoStart from odoEnd✓ (-1 if received as parameters) return answer✓	3	
2.1.4	determineTollFees METHOD: Receive route as string parameter ✓ Determine correct row in 2D depending on routeNr ✓✓ (subtract 1 mark each mistake – max 2) Determine correct column in 2D depending on TruckNr Tr1 OR Tr2: 1 st column✓, Tr3: 2 nd column✓, Tr4 OR Tr5: 3 rd column✓ Find and return tollFees value✓ at position [row] ✓ [column]✓ (-1 for incorrect order)	10	
2.2.1	Button – [Get data from file]: Receive vehicle number from combo box & convert to string✓ {Delphi: AssignFile, Reset Java: Create object to read from file} ✓✓ Test if file does not exist✓ & display message✓ Use loop counter/delivery number of last entry ✓ Increment last delivery number for new Delivery number✓ Loop through file ✓ Read line from text file✓ Find✓ and keep last occurrence of vehicle number✓ Separate information using (#) – to obtain the odometer reading, use split/copy/pos/indexOf ✓ Obtain odometer from string✓ Display values in text boxes✓	14	

QUESTION 2: MARKING GRID - OBJECT-ORIENTED PROGRAMMING - continue

2.2.2	<p>Button – [New delivery]: Obtain newTripNumber, begin odometer, end odometer, truck number ✓ <i>Instantiate the object with values:</i> Left side of assign ✓ Right side of assign, Correct number and order of parameters (4) ✓ and type casting ✓ (<i>or obtain from variables of correct type</i>) Display message after object has been created ✓ Use object ✓ to call calculateDistance ✓ and calculate fuelUsed ✓ (<i>divide by 5</i>) (Java: Ensure the fuelUsed value is a double value) Call set method to set fuelUsed attribute ✓ to calculated value Enable the fuel used and toll fees buttons ✓ (Buttons can also be enabled in 2.2.3)</p>	10	
2.2.3	<p>Button – [Display delivery]: Display information in text area ✓ using toString ✓</p>	2	
2.2.4	<p>Button – [Check fuel used]: Read fuel used from text box and convert to a real number ✓ Use object and call getMethod to get fuel used ✓ Calculate difference ✓, calculate % ✓ Test if less than 10% difference ✓ (provide for positive and negative) ✓ Change fuel used using set method ✓ Display message to indicate change has been made ✓ Else Display error message ✓</p>	9	
2.2.5	<p>Button – [Calculate toll fees]: Read route number from text box ✓ Use object to call determineTollFees method ✓ and send route number as parameter ✓ Display toll fees ✓ in currency format (R##.##) ✓</p>	5	
	TOTAL:	60	

ANNEXURE C:**SECTION C:****QUESTION 3: MARKING GRID – PROBLEM SOLVING**

CENTRE NUMBER:		EXAMINATION NUMBER:	
QUESTION	DESCRIPTION	MAX. MARKS	LEARNER'S MARKS
3.1	<p>Button – [Load item]</p> <p>Declare variable for loading code ✓ Test if fragile is selected ✓ Test if ✓ fragileItems < 20 ✓ Create loading code F ✓ number ✓ Update fragileItems ✓ by adding a * ✓ Else ✓ (Test if non fragile is selected) Test if ✓ nonFragileItems < 30 ✓ Create loading code ✓ Update nonFragileItems by adding a * ✓ Test if there is space for the item ✓ Display loading code in the text box ✓ Else ✓ Display a message indicating the item is not loaded ✓ Display heading ✓ Display string representing fragileItems ✓ Display string representing nonFragileItems ✓</p>	20	
3.2	<p>Button – [Check load status]</p> <p>Determine the number of fragile items Determine the number of non-fragile items ✓ <i>(Mark given for assigning given values 4 and 13) OR</i> <i>(Mark given for using values from 3.1)</i> Calculate percentage fragile items ✓ Calculate percentage non-fragile items ✓ Display column headings ✓ and detail in columns ✓ Display fragile item details, ✓ formatted, percentage to 2 decimal spaces ✓ Display non-fragile item details ✓ % formatted to 2 decimal places If condition <i>(percent fragile >=50 ✓ AND ✓ percent non-fragile >=50 ✓)</i> Nested if acceptable Display message "to progress" on text area ✓ else Display message "may not progress" ✓ If (condition) ✓ (<i>numFragile < 10</i>) Display fragileItems still required on text area, calculation ✓ If (condition) ✓ (<i>numNonFragile < 15</i>) Display nonFragileItems still required on text area, calculation ✓ Note: The condition can be reversed displaying the corresponding messages. Numbers (10 and 15) can be used instead of percentages.</p>	17	

QUESTION 3: MARKING GRID – PROBLEM SOLVING - continue

3.3	Button – [Clear load] Clear fragileItems ✓ Clear nonFragileItems ✓ Clear text area ✓	3	
	TOTAL:	40	

SUMMARY OF LEARNER'S MARKS:

	SECTION A	SECTION B	SECTION C	
	QUESTION 1	QUESTION 2	QUESTION 3	GRAND TOTAL
MAX. MARKS	50	60	40	150
LEARNER'S MARKS				

ANNEXURE D: SOLUTION FOR QUESTION 1: JAVA

```
// A solution to Question 1

package Question1Package;

import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class Question1_Solution extends javax.swing.JFrame {

    int kilometres = 635;
    public Question1_Solution() {
        initComponents();
        this.setLocationRelativeTo(this);
        this.setVisible(true);
        lstKgs.setSelectedIndex(0);
        txfBarCode.setText("639382000393");
    }
    =====
    // Question 1.1
    =====
    private void btnDeliveryActionPerformed(java.awt.event.ActionEvent evt) {
        String departure = (String) (cmbDepart.getSelectedItemAt());
        String destination = (String) (cmbDestination.getSelectedItemAt());
        kilometres = Integer.parseInt(txfDistance.getText());
        lblDelivery.setText(departure + " to " + destination + " : " +
            kilometres + " km");
    }
    =====
    // Question 1.2
    =====
    private void btnDeliveryCostActionPerformed(java.awt.event.ActionEvent evt) {
        int position = (int) (lstKgs.getSelectedIndex());
        double costTransport = 0;
        switch (position) {
            case 0:
                costTransport = 0.6 * kilometres;
                break;
            case 1:
                costTransport = 1.0 * kilometres;
                break;
            case 2:
                costTransport = 1.25 * kilometres;
                break;
            case 3:
                costTransport = 1.65 * kilometres;
                break;
        }

        if (chbSpeedPost.isSelected()) {
            costTransport += 100;
        }
        txfCost.setText(String.format("R%2.2f", costTransport));
    }
}
```

```

=====
// Question 1.3
=====
private void btnBoxNumberActionPerformed(java.awt.event.ActionEvent evt) {
    int boxNumber = 0;
    if (chbSpeedPost.isSelected()) {
        boxNumber = 4;
    } else {
        do {
            boxNumber = (int) (Math.random() * 5) + 1;
        } while (boxNumber == 4);
    }
    txfBoxNumber.setText("" + boxNumber);
}
=====
// Question 1.4
=====
private void btnBarCodeActionPerformed(java.awt.event.ActionEvent evt) {
    String barCode = txfBarCode.getText();
    int sumOdd = 0;
    int sumEven = 0;
    for (int cnt = 0; cnt < barCode.length()-1; cnt ++)
    {
        if ((cnt+1) % 2 ==0)
            sumEven = sumEven + Integer.parseInt(barCode.substring(cnt, cnt + 1));
        else
            sumOdd = sumOdd + Integer.parseInt(barCode.substring(cnt, cnt + 1));
    }
    int sum = sumOdd * 3 + sumEven;
    int checkDigit = 10 - (sum % 10);
    if(checkDigit == Integer.parseInt(barCode.substring(barCode.length()-1)))
    {
        txfDisplayBarCode.setText("The bar code is valid. Check digit: " +
            checkDigit);
    }
    else
    {
        txfDisplayBarCode.setText("The bar code is NOT valid. Correct check
            digit: " + checkDigit);
    }
}
=====
// Question 1.5
=====
private void btnViewDeliveriesActionPerformed(java.awt.event.ActionEvent evt) {
    String place = (String)(cmbCityName.getSelectedItem());
    outputArea.setText(place+"\n");
    try {
        PrintWriter out = new PrintWriter(new FileWriter(
            "December2014"+place+".txt"));
        for (int i = 0;i<arrDecDeliveries.length;i++){
            if(arrDecDeliveries[i].indexOf(place) >=0){
                outputArea.append(arrDecDeliveries[i)+"\n");
                out.println(arrDecDeliveries[i]);
            }
        }
        out.close();
    } catch (IOException e) {
        JOptionPane.showMessageDialog(null,"Error");
    }
}

```


ANNEXURE E: SOLUTION FOR QUESTION 2: JAVA

```
// A solution to Question 2
```

OBJECT CLASS: DELIVERY (GIVEN)

```
public class Delivery {

//=====
//This code is given in the program
//=====
    private int deliveryNum;
    private String truckNum;
    private double fuelUsed;
    private int odoStart;
    private int odoEnd;
    double[][] tollFees = {{105.50, 135.00, 210.00},
                           {35.00, 54.00, 82.00},
                           {85.00, 129.00, 205.00},
                           {112.00, 170.00, 219.00}};

    public String toString() {
        DecimalFormat df = new DecimalFormat("0.0");
        String output = "Delivery number: " + deliveryNum + "\nTruck number: "
            + truckNum + "\nOdometer reading: \n\t(Start) " +
            odoStart + "\n\t(End) " + odoEnd + "\nFuel used: " +
            df.format(fuelUsed) + " litres";

        return output;
    }

//=====
// Question 2.1.1
//=====

    public Delivery(int deliveryNum, String truckNum, int odoStart, int
        odoEnd) {
        this.deliveryNum = deliveryNum;
        this.truckNum = truckNum;
        this.odoStart = odoStart;
        this.odoEnd = odoEnd;
    }

//=====
// Question 2.1.2
//=====
//Accessor method

    public double getFuelUsed() {
        return fuelUsed;
    }

//Mutator method

    public void setFuelUsed(double fuel) {
        fuelUsed = fuel;
    }
}
```

```
=====
// Question 2.1.3
=====
```

```
public int calculateDistance()
{
    return odoEnd - odoStart;
}
```

```
=====
// Question 2.1.4
=====
```

```
public double determineTollFees(String route) {
    double tollAmount = 0;
    int row = Integer.parseInt(route.substring(2, 3)) - 1;

    if (truckNum.equals("Tr1") || truckNum.equals("Tr2")) {
        tollAmount = tollFees[row][0];
    } else if (truckNum.equals("Tr3")) {
        tollAmount = tollFees[row][1];
    } else {
        tollAmount = tollFees[row][2];
    }
    /* Alternative:
    switch (truckNum) {
        case "Tr1":
        case "Tr2":
            tollAmount = tollFees[row][0];
            break;
        case "Tr3":
            tollAmount = tollFees[row][1];
            break;
        default:
            tollAmount = tollFees[row][2];
            break;
    }
    */
    return tollAmount;
}
```

GUI CLASS: QUESTION2_SOLUTION

```
package Question2Package;

import java.io.File;
import java.io.FileReader;
import java.text.DecimalFormat;
import java.util.Scanner;
import javax.swing.JOptionPane;

public class Question2_Solution extends javax.swing.JFrame {
=====
// Given code
=====
Delivery objDelivery;

public Question2_Solution() {
    initComponents();
    this.setLocationRelativeTo(this);
    this.setVisible(true);
    btnTollFees.setEnabled(false);
    btnFuelChange.setEnabled(false);
}

// Code not copied for graphics
=====
// Question 2.2.1
=====
private void btnGetFromFileActionPerformed(java.awt.event.ActionEvent evt) {
    File file = new File("DeliveryInfo.txt");
    if (!file.exists()) {
        JOptionPane.showMessageDialog(rootPane, "File does not exists");
        System.exit(0);
    } else {

        String truckNr = (String) cmbVehicleNumber.getSelectedItem();
        try {
            String lastTruckLine = "", line = "";
            Scanner sc = new Scanner(new FileReader("DeliveryInfo.txt"));
            String[] temp;
            while (sc.hasNext()) {
                line = sc.next();
                if (line.contains(truckNr)) {
                    lastTruckLine = line;
                }
            }
            temp = line.split("#");
            int newTrip = Integer.parseInt(temp[0]) + 1;
            txfNewTripNum.setText("" + newTrip);

            temp = lastTruckLine.split("#");
            txfStartOdometer.setText(temp[2]);
        }
        catch (Exception e) {
        }
    }
}
}
```

```
=====
// Question 2.2.2
=====

private void btnNewDeliveryActionPerformed(java.awt.event.ActionEvent evt) {
    int newTripNum = Integer.parseInt(txfNewTripNum.getText());
    int startOdoReading = Integer.parseInt(txfStartOdometer.getText());
    int endOdoReading = Integer.parseInt(txfEndOdometer.getText());
    String truckNr = (String) cmbVehicleNumber.getSelectedItemAt();
    objDelivery = new Delivery(newTripNum, truckNr, startOdoReading,
        endOdoReading);
    JOptionPane.showMessageDialog(rootPane, "Delivery object created
        successfully.");
    int distance = objDelivery.calculateDistance();
    objDelivery.setFuelUsed(distance / 5.0);
    btnTollFees.setEnabled(true);
    btnFuelChange.setEnabled(true);
}

=====
// Question 2.2.3
=====
private void btnDisplayDeliveryActionPerformed(java.awt.event.ActionEvent evt)
{
    txaOptionA.setText(objDelivery.toString());
}

=====
// Question 2.2.4
=====
private void btnFuelChangeActionPerformed(java.awt.event.ActionEvent evt) {
    double fuelAdded = Double.parseDouble(txfFuel.getText());
    double fuelUsed = objDelivery.getFuelUsed();
    if (Math.abs(fuelAdded - fuelUsed) / fuelUsed < 0.1) {
        objDelivery.setFuelUsed(fuelAdded);
        txfFuelMessage.setText("Fuel changed from " + fuelUsed + " to " +
            fuelAdded + " litres");
    } else {
        txfFuelMessage.setText("ERROR: Difference in fuel used is too great");
    }
}

=====
// Question 2.2.5
=====
private void btnTollFeesActionPerformed(java.awt.event.ActionEvent evt) {
    String output = String.format("%-23sR%2.2f", "Toll fees to be paid:",
        objDelivery.determineTollFees(txfRoute.getText()));
    lblTollFees.setText(output);
}

=====
```

ANNEXURE F: SOLUTION FOR QUESTION 3: JAVA

```
// A possible solution to Question 3

package Question3Package;

import javax.swing.JOptionPane;

public class Q3 extends javax.swing.JFrame {

//Global variables
    String fragileItems = "";
    String nonFragileItems = "";

//=====
//This code is given in the program
//=====

public Question3_Solution() {
    initComponents();
    this.setLocationRelativeTo(this);
    this.setVisible(true);
    rbtFragile.setSelected(true);
}

// Question 3.1
//=====
private void btnLoadActionPerformed(java.awt.event.ActionEvent evt) {
    String loadingCode = "";
    if (rbtFragile.isSelected()) {
        if (fragileItems.length() < 20) {
            loadingCode = "F" + (fragileItems.length() + 1);
            fragileItems += "*";
        }
    } else {
        if (nonFragileItems.length() < 30) {
            loadingCode = "NF" + (nonFragileItems.length() + 1);
            nonFragileItems += "*";
        }
    }
    txfLoadingCode.setText(loadingCode);
    if (loadingCode.equals("")) {
        JOptionPane.showMessageDialog(null, "Loading of item cannot be
        processed - No loading space\n", "Information", WIDTH);
    }
    txaOutput.setText("Loading progress display area:
        \n=====
        \n\n");
    txaOutput.append(String.format("%-20s%-25s\n", "Fragile items:",
        fragileItems));
    txaOutput.append(String.format("%-20s%-25s", "Non-fragile items:",
        nonFragileItems));
}
}
```

```
=====
// Question 3.2
=====

private void btnStatusActionPerformed(java.awt.event.ActionEvent evt) {
    int numFragile = fragileItems.length();
    int numNonFragile = nonFragileItems.length();
    double percFragile = (numFragile) / 20.0 * 100;
    double percNonFragile = (numNonFragile) / 30.0 * 100;
    txaOutput.setText(" Load status report:\n
        =====\n");
    txaOutput.append(String.format("%-15s%-25s%-15s\n", " Item type",
        "Number of items", "Percentage loaded"));
    txaOutput.append(String.format("%-15s%-25s%-13.2f\n", " Fragile",
        numFragile, percFragile));
    txaOutput.append(String.format("%-15s%-25s%-13.2f\n", " Non-fragile",
        numNonFragile, percNonFragile));
    if (percFragile >= 50 && percNonFragile >= 50) {
        txaOutput.append("\n The delivery may progress.");
    }
    if (numFragile < 10 || numNonFragile < 15) {
        txaOutput.append("\n The delivery may not progress.");
        if (numFragile < 10) {
            txaOutput.append("\n Number of fragile items still required : "
                + (10 - numFragile));
        }
        if (numNonFragile < 15) {
            txaOutput.append("\n Number of non-fragile items still
                required : " + (15 - numNonFragile));
        }
    }
}

=====
// Question 3.3
=====

private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {
    fragileItems="";
    nonFragileItems="";
    txaOutput.setText("");
}
}
```

ANNEXURE G: SOLUTION FOR QUESTION 1: DELPHI

```
unit Question1_U_Memo;

interface
    //Possible solution for Question 1

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms,
    Dialogs, StdCtrls, ExtCtrls, ComCtrls, StrUtils;

type
    TfrmQuestion1 = class(TForm)
        grpQ1_1: TGroupBox;
        grpQ1_3: TGroupBox;
        grpQ1_2: TGroupBox;
        grpQ1_4: TGroupBox;
        lblDeliveryFrom: TLabel;
        lblDeliveryTo: TLabel;
        lblNoKM: TLabel;
        edtKm: TEdit;
        btnDeliveryConfirmation: TButton;
        grpDLabel: TGroupBox;
        lblDeliveryCode: TLabel;
        grpSpeedpost: TGroupBox;
        btnDeliveryCost: TButton;
        grpRange: TGroupBox;
        edtDeliveryCost: TEdit;
        lstRangeKM: TListBox;
        btnDeliveryBoxNumber: TButton;
        edtDeliveryBoxNumber: TEdit;
        btnCreateBarCode: TButton;
        edtCreateBarCode: TEdit;
        cboDeliveryFrom: TComboBox;
        cboDeliveryTo: TComboBox;
        lblUPCBarCode: TLabel;
        edtUPCBarCode: TEdit;
        chkSpeedPost: TCheckBox;
        grpQ1_5: TGroupBox;
        cboCityName: TComboBox;
        btnViewDeliveries: TButton;
        redOutputArea: TRichEdit;
        lblCity: TLabel;
        procedure btnDeliveryConfirmationClick(Sender: TObject);
        procedure btnDeliveryCostClick(Sender: TObject);
        procedure FormCreate(Sender: TObject);
        procedure btnDeliveryBoxNumberClick(Sender: TObject);
        procedure btnCreateBarCodeClick(Sender: TObject);
        procedure btnViewDeliveriesClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    frmQuestion1: TfrmQuestion1;
    iKilometres : Integer = 635; //default value
```

```
//given do not change
arrDecDeliveries : array[1..32] of String =
('2013-12-01 Durban to Cape Town',
'2013-12-01 Polokwane to Johannesburg',
'2014-12-02 Cape Town to Johannesburg ',
'2014-12-02 Polokwane to Potchefstroom ',
'2014-12-02 Bloemfontein to Port Elizabeth',
'2013-12-03 Polokwane to Potchefstroom',
'2014-12-03 Cape Town to Port Elizabeth ',
'2014-12-03 Port Elizabeth to Potchefstroom ',
'2014-12-04 Port Elizabeth to Durban',
'2013-12-04 Polokwane to Kimberley',
'2014-12-04 Cape Town to Kimberley ',
'2014-12-04 Polokwane to Potchefstroom ',
'2014-12-04 Kimberley to Port Elizabeth',
'2014-12-05 Durban to Kimberley',
'2014-12-05 Bloemfontein to Potchefstroom',
'2014-12-05 Durban to Potchefstroom',
'2013-12-05 Cape Town to Potchefstroom',
'2013-12-05 Polokwane to Cape Town',
'2014-12-06 Cape Town to Johannesburg ',
'2014-12-06 Polokwane to Potchefstroom ',
'2014-12-06 Bloemfontein to Kimberley',
'2013-12-06 Polokwane to Johannesburg',
'2014-12-07 Cape Town to Port Elizabeth ',
'2014-12-07 Port Elizabeth to Potchefstroom ',
'2014-12-07 Potchefstroom to Durban',
'2013-12-07 Cape Town to Kimberley',
'2014-12-08 Cape Town to Kimberley ',
'2014-12-08 Polokwane to Potchefstroom ',
'2014-12-08 Kimberley to Port Elizabeth',
'2014-12-08 Potchefstroom to Kimberley',
'2014-12-09 Bloemfontein to Polokwane',
'2014-12-09 Durban to Bloemfontein');
```

implementation

```
{ $R *.dfm }
procedure TfrmQuestion1.btnDeliveryConfirmationClick(Sender: TObject);
begin
=====
//Question 1.1
=====
    ikilometres := StrToInt(edtKm.Text);
    lblDeliveryCode.Caption :=
        cboDeliveryFrom.Items[cboDeliveryFrom.ItemIndex] + ' to ' +
        cboDeliveryTo.Items[cboDeliveryTo.ItemIndex] + ' : ' +
        edtKm.Text + 'km';
end;

procedure TfrmQuestion1.btnDeliveryCostClick(Sender: TObject);
var
    iPositon      : integer;
    rCostTransport : real;
begin
=====
//Question 1.2
=====
    iPositon := lstRangeKM.ItemIndex;
    rCostTransport := 0;
    Case (iPositon) of
```



```

0: rCostTransport := 0.60 * ikilometres;
1: rCostTransport := 1.00 * ikilometres;
2: rCostTransport := 1.25 * ikilometres;
3: rCostTransport := 1.65 * ikilometres;
end;

if chkSpeedPost.Checked = True
then
begin
rCostTransport := rCostTransport + 100;
end;
edtDeliveryCost.Text := FloatToStrF(rCostTransport, ffCurrency, 8,2);
end;

procedure TfrmQuestion1.btnDeliveryBoxNumberClick(Sender: TObject);
var
iBoxNumber : integer;
begin
=====
//Question 1.3
=====
if chkSpeedPost.Checked = true
then
begin
iBoxNumber := 4;
end
else
begin
//generate a random number between 1 to 5 which is not 4
repeat
iBoxNumber := random(5)+1;
until iBoxNumber <> 4;
end;
edtDeliveryBoxNumber.Text := IntToStr(iBoxNumber);
end;

procedure TfrmQuestion1.btnCreateBarCodeClick(Sender: TObject);
var
sBarCode : string;
iSumOdd, iSumEven, iCounter, iTotal, iCheckDigit : Integer;
begin
=====
//Question 1.4
=====
sBarCode := edtUPCBarCode.Text;
iSumOdd := 0;
iSumEven := 0;
for iCounter := 1 to Length(sBarCode)-1 do
begin
if (iCounter MOD 2) = 0
then inc(iSumEven, StrToInt(sBarCode[iCounter]))
else inc(iSumOdd, StrToInt(sBarCode[iCounter]));
end;
iTotal := (iSumOdd * 3) + iSumEven;
iCheckDigit := 10 - (iTotal mod 10);
if iCheckDigit = StrToInt(sBarCode[Length(sBarCode)])
then
edtCreateBarCode.Text := 'The bar code is valid. ' +
'Check digit: ' + IntToStr(iCheckDigit)
else
edtCreateBarCode.Text := 'The bar code is NOT valid. ' +

```

```
        IntToStr(iCheckDigit) ;
end;

procedure TfrmQuestion1.btnViewDeliveriesClick(Sender: TObject);
var
    sCity, sFileName      : string;
    txtFile               : TextFile;
    iCounter              : Integer;
begin
    =====
    //Question 1.5
    =====
    redOutputArea.Clear;
    sCity := cboCityName.Items[cboCityName.ItemIndex];
    redOutputArea.Lines.Add(sCity);
    sFileName := 'December2014'+sCity + '.txt';
    AssignFile(txtFile, sFileName);
    Rewrite(txtFile);
    for iCounter := 1 to 32 do
        begin
            if pos(sCity, arrDecDeliveries[iCounter]) > 0
            then
                begin
                    redOutputArea.Lines.Add(arrDecDeliveries[iCounter]);
                    Writeln(txtFile, arrDecDeliveries[iCounter]);
                end;
            end;
        CloseFile(txtFile);
    end;

procedure TfrmQuestion1.FormCreate(Sender: TObject);
begin
    lstRangeKM.Selected[0] := True;
    CurrencyString := 'R';
    Randomize;
end;

end.
```

ANNEXURE H: SOLUTION FOR QUESTION 2: DELPHI**CLASS UNIT: DELIVERY_U.PAS**

```
unit Delivery_U;
  //Possible solution for Question 2 - class unit.

interface

uses
  sysUtils;

type
  TDelivery = class(TObject)
  private
    fDeliveryNum : integer;
    fTruckNum    : string;
    fFuelUsed    : real;
    fOdoStart    : integer;
    fOdoEnd      : integer;
  public
    function toString: string;
    constructor Create(iDeliverNumber: integer; sTruckNumber : string;
                      iOdoStart, iOdoEnd : integer);

    function getFuelUsed: real;
    procedure setFuelUsed (rFuelUsed : Real);
    function calculateDistance: Integer;
    function determineTollFees(sRoute : string): real;
  end;

var
  =====
  //Given to be used in question 2.1.4
  =====
  tollFees : array[1..4,1..3] of real =
    ((105.50, 135.00, 210.00), (35.00, 54.00, 82.00),
     (85.00,129.00,205.00), (112.00, 170.00, 219.00));

implementation
  =====
  // Question 2.1.1.
  =====
  constructor TDelivery.Create(iDeliverNumber: integer; sTruckNumber: string;
    iOdoStart, iOdoEnd: integer);
  begin
    fDeliveryNum := iDeliverNumber;
    fTruckNum    := sTruckNumber;
    fOdoStart    := iOdoStart;
    fOdoEnd      := iOdoEnd;
  end;
  =====
  // Question 2.1.2.
  =====
  function TDelivery.getFuelUsed: real;
  begin
    Result := fFuelUsed;
  end;
```

```

procedure TDelivery.setFuelUsed(rFuelUsed: Real);
begin
  fFuelUsed := rFuelUsed;
end;
=====
// Question 2.1.3.
=====
function TDelivery.calculateDistance: Integer;
begin
  Result := fOdoEnd - fOdoStart;
end;
=====
// Question 2.1.4.
=====
function TDelivery.determineTollFees(sRoute: string): real;
var
  iRow : integer;
begin
  Result := 0;
  iRow := StrToInt(sRoute[3]); //3rd character
  if (fTruckNum = 'Tr1') OR (fTruckNum = 'Tr2')
  then Result := tollFees[iRow, 1]
  else if (fTruckNum = 'Tr3')
  then Result := tollFees[iRow, 2]
  else Result := tollFees[iRow, 3];

  {Alternative:
  case fTruckNum[3] of
    '1', '2' : Result := tollFees[iRow, 1];
    '3'      : Result := tollFees[iRow, 2];
    '4', '5' : Result := tollFees[iRow, 3];
  end; //case
  }
end;

function TDelivery.toString: string;
begin
  Result := 'Delivery Number: ' + IntToStr(fDeliveryNum) + #13 +
    'Truck number: ' + fTruckNum + #13 +
    'Odometer reading: '+#13+
    #9 + '(Start) ' + IntToStr(fOdoStart) + #13 +
    #9 + '(End)   ' + IntToStr(fOdoEnd) + #13 +
    'Fuel used: ' + FloatToStr(fFuelUsed) + ' litres';
end;

end.

```

MAIN FORM UNIT: QUESTION2_U.PAS

```
unit Question2_U_Memo;
//Possible solution for Question 2 - Formunit.

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, ExtCtrls, StdCtrls, Spin, Delivery_U, ComCtrls;

type
  TfrmQuestion2 = class(TForm)
    pnlTitle: TPanel;
    grpCreatedisplay: TGroupBox;
    GroupBoxOptionC: TGroupBox;
    lblVehicleNumber: TLabel;
    cboVehicleNumber: TComboBox;
    redOutput: TRichEdit;
    lblDistanceTravelled: TLabel;
    edtEndOdometer: TEdit;
    btnCreateNewDelivery: TButton;
    btnShowDelivery: TButton;
    lblActualFuelUsed: TLabel;
    edtFuelUsed: TEdit;
    btnFuelChange: TButton;
    edtFuelMessage: TEdit;
    grpTollFees: TGroupBox;
    lblRoute: TLabel;
    edtRoute: TEdit;
    btnTollFee: TButton;
    pnlTollFees: TPanel;
    lblStartOdoReading: TLabel;
    edtStartOdometer: TEdit;
    lblNewTripNum: TLabel;
    edtNewTripNum: TEdit;
    btnGetFromFile: TButton;
    lblTollFees: TLabel;

    procedure FormCreate(Sender: TObject);
    procedure btnTollFeeClick(Sender: TObject);
    procedure btnCreateNewDeliveryClick(Sender: TObject);
    procedure btnShowDeliveryClick(Sender: TObject);
    procedure btnFuelChangeClick(Sender: TObject);
    procedure btnGetFromFileClick(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion2: TfrmQuestion2;

  Delivery : TDelivery;

implementation
```

```
{ $R *.dfm }
{ $R+ }

procedure TfrmQuestion2.FormCreate(Sender: TObject);
begin
  CurrencyString := 'R';
end;

procedure TfrmQuestion2.btnGetFromFileClick(Sender: TObject);
var
  txtFile : TextFile;
  sLine, sTripNo, sTruckNumber, sStartOdo : string;
  iNewTrip : Integer;
begin
  =====
  // Question 2.2.1
  =====
  if NOT FileExists('DeliveryInfo.txt')
  then
  begin
    MessageDlg('DeliveryInfo.txt does not exist', mtError, [mbOK], 0);
    Exit;
  end;

  sTruckNumber := cboVehicleNumber.Items[cboVehicleNumber.ItemIndex];

  AssignFile(txtFile, 'DeliveryInfo.txt');
  Reset(txtFile);
  while not EOF(txtFile) do
  begin
    readln(txtFile, sline);
    sTripNo := copy(sline, 1, pos('#', sline)-1);
    if Pos(sTruckNumber, sLine) > 0
    then
    begin
      Delete(sline, 1, pos('#', sline));
      Delete(sline, 1, pos('#', sline));
      sStartOdo := sLine;
    end; //if
  end;
  closeFile(txtFile);

  iNewTrip := StrToInt(sTripNo) + 1;
  edtNewTripNum.Text := IntToStr(iNewTrip);
  edtStartOdometer.Text := sStartOdo;
end;

procedure TfrmQuestion2.btnCreateNewDeliveryClick(Sender: TObject);
var
  iDistance, iNewTripNum, iStartOdoReading, iEndOdoReading : integer;
  sTruckNumber : string;
begin
  =====
  // Question 2.2.2
  =====
  sTruckNumber := cboVehicleNumber.Items[cboVehicleNumber.ItemIndex];
  iNewTripNum := StrToInt(edtNewTripNum.text);
  iStartOdoReading := StrToInt(edtStartOdometer.text);
  iEndOdoReading := StrToInt(edtEndOdometer.text);
```

```

Delivery := TDelivery.Create(iNewTripNum, sTruckNumber, iStartOdoReading,
                             iEndOdoReading);
MessageDlg('Delivery object created successfully.', mtInformation,
           [mbOK],0);

iDistance := Delivery.calculateDistance;
Delivery.setFuelUsed(iDistance / 5.0);

btnTollFee.Enabled := True;
btnFuelChange.Enabled := True;
end;

procedure TfrmQuestion2.btnShowDeliveryClick(Sender: TObject);
begin
=====
// Question 2.2.3
=====
  redOutput.Clear;
  redOutput.Lines.Add(Delivery.toString);
end;

procedure TfrmQuestion2.btnFuelChangeClick(Sender: TObject);
var
  rFuelAdded, rFuelUsed : real;
begin
=====
// Question 2.2.4
=====
  rFuelAdded := StrToFloat( edtFuelUsed.Text);
  rFuelUsed := delivery.getFuelUsed;

  if (Abs(rFuelAdded - rFuelUsed) / rFuelUsed) < 0.1
  then
    begin
      Delivery.setFuelUsed(rFuelAdded);
      edtFuelMessage.Text := 'Fuel used changed from ' +
        FloatToStrF(rFuelUsed, ffFixed, 12, 1) + ' to ' +
        FloatToStrF(rFuelAdded, ffFixed, 12, 1) + ' litres';
    end
  else
    edtFuelMessage.Text := 'ERROR: : Difference in fuel used is too great';
end;

procedure TfrmQuestion2.btnTollFeeClick(Sender: TObject);
var
  sRouteNum : string;
begin
=====
// Question 2.2.5.
=====
  sRouteNum := edtRoute.Text;
  lblTollFees.Caption := 'Toll fees to be paid: ' +
    FloatToStrF(Delivery.determineTollFees(sRouteNum), ffCurrency, 8, 2);
end;

end.

```

ANNEXURE I: SOLUTION FOR QUESTION 3: DELPHI

```

unit Question3_U_Memo;
//Possible solution for Question 3.

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, ExtCtrls, StdCtrls, ComCtrls;

type
  TfrmQuestion3 = class(TForm)
    grpLoadingZone: TGroupBox;
    btnClearLoad: TButton;
    btnLoadItem: TButton;
    btnCheckLoadingStatus: TButton;
    rgpItemType: TRadioGroup;
    redQ3: TRichEdit;
    edtLoadingCode: TEdit;
    lblLoadingCode: TLabel;
    procedure btnClearLoadClick(Sender: TObject);
    procedure btnLoadItemClick(Sender: TObject);
    procedure btnCheckLoadingStatusClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion3: TfrmQuestion3;

  sFragileItems      : string;
  sNonFragileItems  : string;

implementation
{$R *.dfm}
{$R+}

procedure TfrmQuestion3.btnLoadItemClick(Sender: TObject);
var
  sLoadingCode : string;
begin
  =====
  // Question 3.1
  =====
  case rgpItemType.ItemIndex of
    0 : begin
        if length(sFragileItems) < 20
        then
          begin
            sLoadingCode := 'F' + IntToStr(Length(sFragileItems)+1);
            sFragileItems := sFragileItems + '*';
          end
        else sLoadingCode := '';
        end;//fragile
  end;
end;

```



```

1 : begin
  if length(sNonFragileItems) < 30
  then
    begin
      sLoadingCode := 'NF' + IntToStr(Length(sNonFragileItems)+1);
      sNonFragileItems := sNonFragileItems + '*';
    end
  else sLoadingCode := '';
  end;//non-fragile
end;//case

if sLoadingCode = ''
then
  begin
    edtLoadingCode.Clear;
    MessageDlg('Loading of item cannot be processed - No loading space',
              mtInformation, [mbok], 0);
  end;//if no space
else
  begin
    edtLoadingCode.Text := sLoadingCode;
    redQ3.Clear;
    redQ3.Paragraph.TabCount := 1;
    redQ3.Paragraph.Tab[0] := 150;
    redQ3.Lines.Add('Loading progress display area:');
    redQ3.Lines.Add('=====');
    redQ3.Lines.Add(' ');
    redQ3.Lines.Add('Fragile items:' + #9 + sFragileItems);
    redQ3.Lines.Add('Non-fragile items:' + #9 + sNonFragileItems);
  end;//space available
end;

procedure TfrmQuestion3.btnCheckLoadingStatusClick(Sender: TObject);
var
  iNumFragile, iNumNonFragile : integer;
  rPecFragile, rPercNonFragile : real;
begin
  =====
  // Question 3.2
  =====
  iNumFragile := Length(sFragileItems);
  iNumNonFragile := Length(sNonFragileItems);
  rPecFragile := iNumFragile / 20 * 100;
  rPercNonFragile := iNumNonFragile / 30 * 100;
  redQ3.Clear;
  redQ3.Paragraph.TabCount := 2;
  redQ3.Paragraph.Tab[0] := 100;
  redQ3.Paragraph.Tab[1] := 275;
  redQ3.Lines.Add('Load status report:');
  redQ3.Lines.Add('=====');
  redQ3.Lines.Add(' ');
  redQ3.Lines.Add('Item type' + #9 + 'Number of items' + #9 + 'Percentage
    loaded');
  redQ3.Lines.Add('Fragile' + #9 + IntToStr(iNumFragile) + #9 +
    FloatToStrF(rPecFragile, ffFixed, 8,2));
  redQ3.Lines.Add('Non-fragile' + #9 + IntToStr(iNumNonFragile) + #9 +
    FloatToStrF(rPercNonFragile, ffFixed, 8,2));
  redQ3.Lines.Add(' ');

```

```
if (rPecFragile >= 50) and (rPercNonFragile >= 50)
then
begin
redQ3.Lines.Add('The delivery may progress.');
```

```
end
else
begin
redQ3.Lines.Add('The delivery may not progress.');
```

```
if(iNumFragile <= 10 )
then redQ3.Lines.Add('Number of fragile items still required: ' +
IntToStr(10 - iNumFragile) );

if (iNumNonFragile <= 15 )
then redQ3.Lines.Add('Number of non-fragile items still required: '
+ IntToStr(15 - iNumNonFragile));
end;
end;

procedure TfrmQuestion3.btnClearLoadClick(Sender: TObject);
begin
=====
// Question 3.3
=====
sFragileItems := '';
sNonFragileItems := '';
redQ3.Clear;
end;

procedure TfrmQuestion3.FormCreate(Sender: TObject);
begin
CurrencyString := 'R';
end;

end.
```



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

INFORMATION TECHNOLOGY P2

NOVEMBER 2014

MARKS: 150

TIME: 3 hours

This question paper consists of 16 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper consists of SIX sections:

SECTION A:	Multiple-choice Questions	(10)
SECTION B:	System Technologies	(25)
SECTION C:	Communication and Network Technologies	(25)
SECTION D:	Data and Information Management	(25)
SECTION E:	Solution Development	(25)
SECTION F:	Integrated Scenario	(40)

2. Answer ALL the questions.
3. Read ALL the questions carefully.
4. The mark allocation generally gives an indication of the number of facts required in your answer.
5. Number the answers correctly according to the numbering system used in this question paper.
6. Write neatly and legibly.

SECTION A: MULTIPLE-CHOICE QUESTIONS**QUESTION 1**

Various options are given as possible answers to the following questions. Choose the answer and write only the letter (A–D) next to the question number (1.1–1.10) in the ANSWER BOOK.

- 1.1 Which ONE of the following is NOT an example of an operating system?
- A Microsoft Windows 8
 - B Microsoft Office
 - C iOS
 - D Ubuntu (1)
- 1.2 Which ONE of the following devices is needed to allow communication between networks?
- A Router
 - B Switch
 - C Hub
 - D Hot Spot (1)
- 1.3 Convergence refers to ...
- A technology that is always on.
 - B technology that does not need to be tethered to wired connections.
 - C functionalities of many devices built into one.
 - D the ability of mobile devices to connect to one another. (1)
- 1.4 A Trojan horse is ...
- A a malicious program disguised as an innocent application.
 - B software that tries to monitor and track which applications you use on your computer.
 - C software that downloads and displays pop-up adverts when your computer is connected to the Internet.
 - D an example of application software. (1)
- 1.5 The ... translates an entire program into machine language before execution.
- A compiler
 - B interpreter
 - C translator
 - D operating system (1)
- 1.6 Which ONE of the following statements is NOT true?
- A An interactive web page might contain code that can be executed.
 - B Flash-applets might be part of a static web page.
 - C A static web page can be interactive.
 - D Facebook is a dynamic website. (1)

- 1.7 A plug-in is ...
- A system software which controls all activities that take place in a computer.
 - B software which adds functionality to a program.
 - C software which allows the operating system to communicate with and control a hardware device.
 - D an area of storage that the operating system reserves for its own use. (1)
- 1.8 An organisation's private network environment consisting of documents and resources relating to the organisation's business is known as their ...
- A Ethernet.
 - B extranet.
 - C intranet.
 - D Internet. (1)
- 1.9 The BIOS is responsible for ...
- (i) primary memory management.
 - (ii) controlling hardware.
 - (iii) all logical processing in a computer.
 - (iv) locating and loading the operating system.
- A (i) and (ii)
 - B (ii) and (iv)
 - C (iii) and (iv)
 - D (i) and (iii) (1)
- 1.10 The following algorithm tests whether a value is greater than 50 and less than or equal to 60. The algorithm is tested with a number of input values.
- Which set of input values for the variable **Size** will cause the final value of the variable **InRange** to be set to True?
- | |
|--|
| <pre>Line 1 Loop 2 Size ← Read Number 3 InRange ← False 4 If Size > 50 And Size <= 60 Then 5 InRange ← True</pre> |
|--|
- A 49, 53, 50
 - B 51, 57, 61
 - C 50, 54, 60
 - D 51, 55, 59 (1)

TOTAL SECTION A: 10

SECTION B: SYSTEM TECHNOLOGIES**QUESTION 2**

Your school is doing an audit of the technology in the school. Decisions about upgrading and replacing various technologies will be based on the outcome of this audit.

- 2.1 As part of the audit process the hardware in the computer laboratory was checked. It has been suggested that the RAM of the computers in the laboratory be upgraded.
- 2.1.1 What is the purpose of RAM? (2)
- 2.1.2 Give TWO reasons why it may not be possible to upgrade the RAM of old computers. (2)
- 2.2 Mobile devices are used at the school for research purposes. The firmware of mobile devices is stored in read-only memory (ROM).
- 2.2.1 Give ONE reason why ROM is used to store firmware. (1)
- 2.2.2 Give TWO reasons why it is beneficial to update firmware on a mobile device. (2)
- 2.2.3 The school has been warned against jail breaking of mobile devices.
- Explain what *jail breaking* means in this context. (2)
- 2.3 Your school would like to use educational applications that have been developed for mobile devices and desktop computers.
- 2.3.1 High-level languages are commonly used to develop software applications.
- What are the advantages when a programmer uses a high-level programming language instead of a low-level programming language? (2)
- 2.3.2 State ONE challenge faced by developers when they design software applications for mobile devices. (1)
- 2.3.3 Some software applications are classified as open-source software.
- Explain what is meant by *open-source software*. (2)

- 2.4 The Internet is used by learners to do research and this can be time-consuming.
- 2.4.1 Describe how web caching speeds up the web browsing process. (2)
- 2.4.2 Describe another type of caching that may enhance the performance of a computer system. (2)
- 2.5 Your school wants to use the cloud to store administrative data.
- 2.5.1 Explain what is meant by *storing data in the cloud*. (2)
- 2.5.2 Motivate why it could be problematic if the school only uses the cloud to store their administrative data.
- State TWO problems that may occur as part of your motivation. (2)
- 2.6 To encourage interactive learning, wearable technology such as interactive glasses is being investigated.
- 2.6.1 State ONE advantage of interactive glasses. (1)
- 2.6.2 Name ONE type of technology that is used in interactive glasses. (1)
- 2.6.3 Why do you think the use of interactive glasses has been banned in some workplaces? (1)
- TOTAL SECTION B: 25**

SECTION C: COMMUNICATION AND NETWORK TECHNOLOGIES**QUESTION 3**

During the audit process it was established that your school uses a cabled and a wireless network to share hardware resources.

- 3.1 Some aspects need to be clarified for the auditors.
- 3.1.1 Give TWO reasons for having a network other than the sharing of hardware resources. (2)
 - 3.1.2 State TWO benefits of using a star topology for the cabled part of the network. (2)
 - 3.1.3 Give TWO reasons why fibre-optic cable is not the best option for a school network. (2)
 - 3.1.4 Briefly describe the role of an access point in a wireless network. (1)
- 3.2 VoIP is one of the protocols that is used by the school network.
- Motivate the need for VoIP in an interactive learning environment. (2)
- 3.3 Your school uses an ADSL Internet connection.
- 3.3.1 Give TWO reasons why ADSL can be considered to be the best Internet connection option for the school. (2)
 - 3.3.2 Name any other readily available Internet connection that may be used by the school. (1)
 - 3.3.3 A firewall prevents unauthorised online access from outside the school network.

Suggest TWO other measures that could be implemented to prevent unauthorised online access. (2)
- 3.4 Many teachers choose to use their personal mobile devices to send and receive e-mail on the school's wireless network.
- 3.4.1 Push technology is used to send and receive e-mails on a mobile device. Explain what *push technology* is. (2)
 - 3.4.2 E-mail attachments are usually not automatically downloaded on mobile devices.

Give TWO reasons why this is the case. (2)

- 3.5 Teachers are encouraged to create podcasts of their lessons to promote e-learning.
What is a *podcast*? (2)
- 3.6 Some learners use GPS technology in their school projects.
- 3.6.1 What does a GPS device use to determine a specific location? (1)
- 3.6.2 Give TWO examples of how GPS technology can be used by learners as part of school projects. (2)
- 3.7 Learners create personal area networks in the classroom to do their research.
State TWO useful applications of a personal area network while doing research. (2)
- TOTAL SECTION C: 25**


SECTION D: DATA AND INFORMATION MANAGEMENT**QUESTION 4**



The audit process showed that a database needs to be designed and implemented to store data for the tuck-shop.

- 4.1 Quality data is required as input to the database.
- 4.1.1 Briefly describe the difference between *data* and *information*. (2)
- 4.1.2 Name and briefly describe THREE characteristics of quality data. (3)
- 4.2 Your school has advertised a position for a database administrator.
- 4.2.1 State TWO responsibilities of a database administrator. (2)
- 4.2.2 Applicants for the database administrator post need to have some knowledge of SQL injection attacks.
- Briefly explain how SQL injection can be used as a hacking tool. (2)

- 4.3 It has been suggested that the **Tuckshop** database should contain two tables – one table, called **SuppliersTb**, must contain data about the suppliers and another, called **OrdersTb**, must contain data about all the items ordered from the suppliers.

The suggested table structures are as follows:

SuppliersTb		
Key	Field name	Data type
	SupplierID	AutoNumber
	Name	Text
	ContactNumber	Text

OrdersTb		
Key	Field name	Data type
	ItemName	Text
	SupplierID	Number
	UnitPrice	Number
	DateOrdered	Date/Time
	QuantityOrdered	Number
	DeliveryDate	Date/Time
	DeliveryCost	Currency

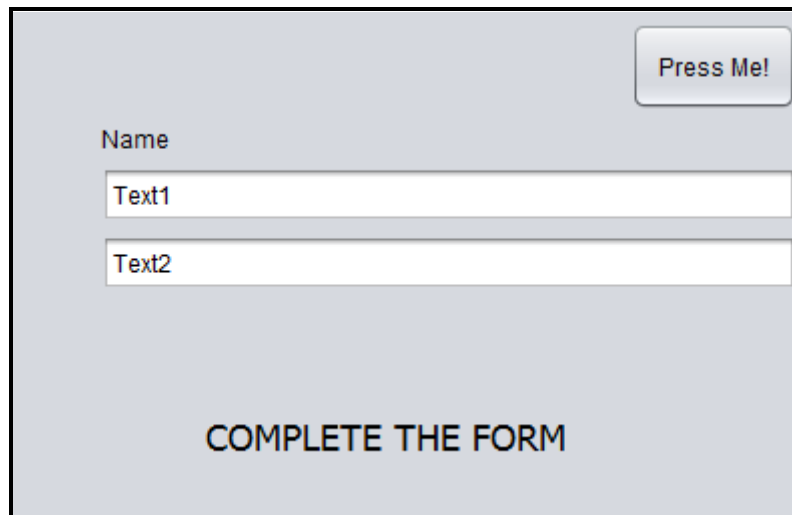
- 4.3.1 Briefly explain why the suggested **Tuckshop** database is a relational database. (1)
- 4.3.2 Use the suggested table structures and write SQL statements for each of the following queries:
- (a) Display the content of the **OrdersTb** (all the fields) in descending order of the quantity of items ordered. (3)
- (b) For each item, display the name of the item, the unit price, the quantity ordered and the total cost for the order in a new field named **TotalCost**. (3)
- NOTE:** Total cost is a calculated field and is calculated by multiplying the unit price by the quantity that was ordered. (3)
- (c) Increase the unit price for all records in the **OrdersTb** table by 5%. (3)
- (d) Remove all records from the **OrdersTb** where the order date is before 1/1/2013. (2)
- (e) Display the supplier name, contact number and item name for all deliveries that will arrive in 2015. (4)

TOTAL SECTION D: 25

SECTION E: SOLUTION DEVELOPMENT**QUESTION 5**

The auditors suggested that new software should be developed to manage learner information.

- 5.1 The following GUI is currently used to allow a user to input learner information.



- 5.1.1 Comment critically on THREE design aspects of the given GUI that need to be improved on in the new system. (3)

- 5.1.2 Identify a component on the given GUI and an event of that component that can be utilised in an event-driven solution. (2)

- 5.2 Data validation is an important technique used to ensure that input data is valid.

A user is required to input a FOUR-character security code which contains ONE alphabetical character and THREE digits. The first character must be alphabetical, for example D845.

Write an algorithm using pseudocode to validate the code for the correct format once the code has been entered.

- NOTE:** Use at least ONE loop as part of your solution. (3)

5.3 Object-orientated programming (OOP) will be used in the design of the new software.

5.3.1 Explain the concept of *encapsulation* in object-orientated programming. (1)

5.3.2 A design is needed for a **Learner** class for the school's administrative program.

Draw a unified modelling language (UML) diagram for an object class called **Learner** and include the following:

- Name and Age attributes with appropriate data types
- Constructor without parameters
- Mutator method for the Name attribute

NOTE: Also indicate the accessibility of the attributes and methods. (5)

5.4 A security feature of the new software will use a check digit. The digit is generated by counting the number of factors of the user code.

A factor of a given number is a number which can be divided into the given number without a remainder.

The algorithm below, written in pseudocode, determines the number of factors that a given number has.

```

Line
1   Number ← 4
2   Count ← 1
3   NumFactors ← 0
4   While Count <= Number
5       If Number modulus Count = 0 Then
6           NumFactors ← NumFactors + 1
7           Count ← Count + 1
8   Display NumFactors
    
```

NOTE: The result of the modulus operation is the remainder when a number is divided by another number.

Use the following headings to draw and complete a trace table in your ANSWER BOOK to show how the content of the variables change while tracing the steps of the given algorithm. Use as many lines as you require.

Number	Count	NumFactors	Count <= Number ?	Number modulus Count = 0 ?	Display

(6)

- 5.5 An algorithm needs to be designed for a program to manage a learner's marks. You have been provided with a line of pseudocode below which populates an array with marks.

Write an algorithm using pseudocode to sort the array of marks into descending order.

`arrMarks ← (67, 65, 78, 83, 67, 72, 65)` (5)

TOTAL SECTION E: 25

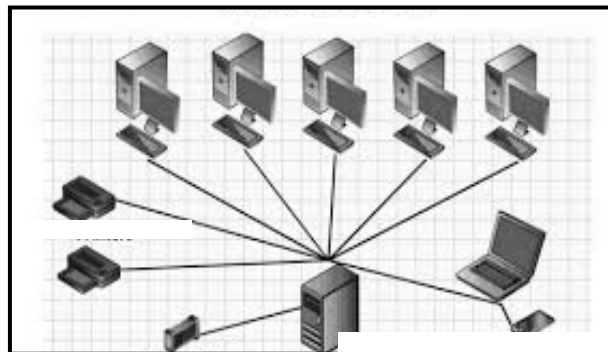
SECTION F: INTEGRATED SCENARIO**QUESTION 6**

Read the scenario below and answer the questions that follow.

SCENARIO

After the implementation of the suggestions from the audit, your school is recognised as one of the leading schools in terms of the use of ICT. The school has an e-learning portal where teachers can upload learning materials and learners can access these materials on an interactive basis. An IT technician has been appointed to assist with the new technology.

- 6.1 The different departments in the school are all networked and use a central server as shown in the diagram below.



- 6.1.1 A client-server network is being currently used.
Give TWO reasons why a client-server network would be a better option than a peer-to-peer network. (2)
- 6.1.2 State TWO hardware features of a server that are significantly different from those of a desktop computer. (2)
- 6.1.3 The school server uses virtualisation to run different servers on a single physical server.
(a) Give ONE example of virtualisation software. (1)
(b) Explain how different operating systems can be used in different virtual servers on the same physical server. (2)

- 6.2 The school runs a dynamic website which is often updated with news and photographs of school events.
- 6.2.1 The school's website offers an RSS feed.
Give TWO reasons why an RSS feed is useful. (2)
- 6.2.2 The website uses lossy compression for the photographs that are uploaded.
Explain the concept *lossy compression*. (1)
- 6.2.3 Many learners and parents view the school's website on their mobile devices.
In what TWO ways could the mobile version of the website differ from the standard version? (2)
- 6.2.4 The principal has learnt that Web 3.0 is the latest evolution of websites.
Why would a school prefer a Web 3.0 site to a Web 2.0 site? (2)
- 6.3 Parents are able to pay school fees online through the school website. The school website uses encryption to ensure that parent details are secure.
- 6.3.1 How can parents ascertain that the school website is secure from the information on their web browsers? (1)
- 6.3.2 Briefly describe what *encryption* is. (2)
- 6.3.3 Parents are encouraged to use 'strong' passwords.
Give TWO guidelines parents can follow to create strong passwords. (2)
- 6.4 The school has joined a social media website.
- 6.4.1 Name TWO ways in which the school will benefit from having a presence on a social media website. (2)
- 6.4.2 Why would the school warn learners about posting personal information, such as phone numbers and addresses, on their profiles? (2)
- 6.4.3 State TWO ways in which the school can ensure its acceptable use policy (AUP) will be implemented when the social media website is used. (2)

- 6.5 The majority of teachers are now using e-learning to enhance their teaching and learning.
- 6.5.1 State TWO ways in which e-learning may be beneficial to the learner. (2)
- 6.5.2 Lesson videos are made available to download or stream.
What is difference between *downloading* and *streaming*? (2)
- 6.5.3 Your school is planning live broadcasts for Mathematics revision lessons.
Give TWO requirements of connectivity to successfully host a live broadcast. (2)
- 6.6 Learners and teachers are encouraged to use mobile devices such as smartphones and tablets for teaching and learning.
- 6.6.1 Name TWO ways in which mobile technology can improve teaching and learning. (2)
- 6.6.2 Mobile devices are powered by rechargeable batteries.
State TWO techniques that are used to extend the battery life of these devices. (2)
- 6.7 The school's IT technician telecommutes on specific days. Telecommuting is using technology to work from home instead of actually being present in the workplace.
- 6.7.1 Name TWO tools/technologies that make telecommuting possible. (2)
- 6.7.2 Identify ONE possible challenge with telecommuting that is not technology-related. (1)
- 6.8 Your school is investigating the option of using Software as a Service (SaaS) to provide access to applications for the development of e-learning resources.
Motivate, by giving TWO reasons, why it would be advantageous for your school to use SaaS. (2)

TOTAL SECTION F: 40
GRAND TOTAL: 150



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

INFORMATION TECHNOLOGY P2

NOVEMBER 2014

MEMORANDUM

MARKS: 150

This memorandum consists of 17 pages.

SECTION A: MULTIPLE-CHOICE QUESTIONS**QUESTION 1**

- | | | |
|------|----------|-----|
| 1.1 | B ✓ | (1) |
| 1.2 | A ✓ | (1) |
| 1.3 | C ✓ | (1) |
| 1.4 | A ✓ | (1) |
| 1.5 | A ✓ | (1) |
| 1.6 | C ✓ | (1) |
| 1.7 | B ✓ | (1) |
| 1.8 | C ✓ | (1) |
| 1.9 | B ✓ | (1) |
| 1.10 | C or D ✓ | (1) |

TOTAL SECTION A: 10

SECTION B: SYSTEM TECHNOLOGIES**QUESTION 2**

- 2.1.1 Temporarily storing data and instructions ✓ to be processed before the CPU needs to work with this data. ✓
Concepts:
Storage
Processing/working with (2)
- 2.1.2 Any TWO of: ✓✓
- The amount of memory that can be used by the specific operating system used in the laboratory may not be able to use the upgraded amount of memory.
 - The RAM slots on the motherboard may be full/No space on the motherboard.
 - New RAM may not be compatible with the motherboard/Not the correct type/Not manufactured anymore. (2)
- 2.2.1 Any ONE of: ✓
- Non-volatile
 - Fast memory to load the firmware
 - Prevents accidental changing of firmware (1)
- 2.2.2 Any TWO of: ✓✓
- Resolve bugs in the operating firmware
 - Add features for the device
 - Improve the performance of the operating system
 - Make provision for newer hardware
 - Make provision for newer software (2)
- 2.2.3 *Jail breaking* is the process of loading a pirate/cracked/unofficial firmware/operating system ✓ onto the mobile device to add/allow features/functionality not available/allowed ✓ with the original firmware.
OR
Stripping a mobile device of certain programmed limitations in order to do things like buy, install and run third party applications that have not been approved or even change the device's processing speed.
Concepts
Software cracking/changing/replacing
Add features previously not possible
For 1 mark: changing the software/firmware of the device (2)
- 2.3.1
- Instructions are written in a language that is easier to understand/simpler code. ✓
 - Libraries of pre-written code are supplied so that complex instructions can be written in a few lines of code/fewer program lines needed/easier to program. ✓ (2)

- 2.3.2 *Any ONE of:* ✓
- Small display area
 - Program must respond quickly/limited CPU or RAM
 - Program must be able to work on different platforms/operating systems/compatible software
 - Should be able to work with limited storage space/use less data
 - Keep power saving in mind
- (1)
- 2.3.3 *Any TWO of* ✓✓
- Software that is free to use
 - and for which the source code is available
 - to be changed or adapted.
- (2)
- 2.4.1 Recently accessed web pages are stored ✓ locally on the hard drive so that they can be retrieved quicker the next time they are visited. ✓
Do not accept: pre-emptive
- (2)
- 2.4.2 *Any ONE of:* ✓✓
- CPU - modern CPUs run at speeds much higher than the motherboard. They need to make extensive use of cache which stores recently used instructions and instructions likely to be used again to improve processing speed.
 - Hard disk cache – RAM is used to speed up access to slower storage devices by storing data that is required in faster RAM instead of accessing the slower storage every time data is required.
- Mark Allocation:
Type of caching
Description
- (2)
- 2.5.1 Data is stored on remote servers/service/computer/hard drive ✓
✓ on the Internet ✓ rather than on a local server or personal computer.
For 1 mark: Online storage
- (2)
- 2.5.2 *Any TWO of:* ✓✓
- Access to data is only possible if you have a good internet access.
 - Online threats to security of data.
 - If remote servers fail then there is no backup.
 - Cloud computing services are only available if subscription fees are paid/data costs.
 - When server/service is down, no access to your data
- (2)
- 2.6.1 *Any ONE of:* ✓
- You can interact with the technology hands free/wearable/ease of use.
 - Interactively provide information on your environment/continuous access to the Internet.
 - Any other acceptable answer
 - Do not accept: synchronisation
- (1)

- 2.6.2 *Any ONE of:* ✓
- Holographic/projection technology
 - Bluetooth
 - GPS
 - Cellular/3G
 - WiFi
 - Google Glass
- (1)

- 2.6.3 *Any ONE of:* ✓
- It is illegal to video or audio record people or aspects such as their performance without their consent. ✓
 - Distraction in workplace
 - Illegal use in tests
 - Security issues
 - Any other valid example
- (1)

TOTAL SECTION B: 25

SECTION C: COMMUNICATION AND NETWORK TECHNOLOGIES**QUESTION 3**3.1.1 *Any TWO of ✓✓*

- Fast, efficient communication for staff and learners.
- Centralisation of data.
- Centralised security
- Transfer of files.
- Sharing software.
- Leisure.
- Flexible access.
- Share a common Internet connection/account.

Do not accept : anything to do with hardware

(2)

3.1.2 *Any TWO of ✓✓*

- Easy to troubleshoot problems.
- Traffic is sent directly to receiver/network is more efficient/high speed.
- If there is a cable break, only that device is disconnected from the network.
- Easy to add/remove devices.
- Easy to set up
- Stable network

(2)

3.1.3 *Any TWO of ✓✓*

- Costly
- Needs an expert to install
- Short distances therefore no need for long range cables
- Limited number of users so no need for large bandwidth
- Lack of flexibility/break easily

(2)

3.1.4 To allow wireless devices to connect to the network ✓ /distribution point for network/send and receive signals.

Do not accept: Hotspot

(1)

3.2 VoIP – Transfer video and voice ✓ for real time communication. ✓

Concepts:

Understanding VoIP

Practical reason for using it in the learning environment.

OR

2 examples of using VoIP

For 1 mark: Expansion of the acronym

Do not accept: Interactivity

(2)

- 3.3.1 *Any TWO of:* ✓✓
- Permanent connection.
 - High speed Internet access.
 - Cost effective
 - Scalable
- (2)
- 3.3.2 *Any ONE of* ✓
- WiMAX/iBurst
 - Leased line
 - Cellular connection (3G/LTE/4G)
 - Satellite
- Don't accept: Wifi
- (1)
- 3.3.3 *Any TWO of:* ✓✓
- Restriction on log in times/time frame to log in.
 - Passwords/required to log in.
 - Anti spyware/malware/anti-virus.
 - Training of personnel.
 - Physical limitations e.g. unplugging connection.
 - Limiting specific IP addresses.
 - Limit the sites that can be visited.
 - Any other valid reason.
- (2)
- 3.4.1 *Push* technology is when the server notifies the device ✓ that a new message has arrived and then downloads it to the mobile device. ✓
- Concepts:
Automated download
FOR 1 mark: Notification only
- (2)
- 3.4.2 *Any TWO of:* ✓✓
- To reduce data usage as mobile data is expensive.
 - Small storage area.
 - Application may not be installed to open/read the attachment/cannot handle format of document.
- (2)
- 3.5 A podcast is an audio/video/media file ✓ that is made available for download/store on your device ✓ from the Internet.
- (2)
- 3.6.1 *Any ONE of:* ✓
- GPS receives signals from GPS satellite orbiting the earth.
 - A valid example of a device that uses GPS, e.g. Cell phone
 - Any description of the method to determine location.
- (1)
- 3.6.2 *Any TWO of:* ✓✓
- Geo-tagging - recording physical location of the project area.
 - To show where a photo was taken for the project.
 - To give coordinates for learners to find a location.
 - Verification that project was conducted at correct location.
 - Any other correct suggestion.
- (2)

3.7 *Any TWO of:* ✓✓

- Synchronising personal devices.
- Transferring data between devices.
- Connections can be made to other people in close proximity.
- Connect to hardware such as a printer, scanner or Internet modem.

Concepts:

Transfer of data

Practical example

**TOTAL SECTION B: (2)
25**

SECTION D: DATA AND INFORMATION MANAGEMENT

- 4.1.1
- Data refers to raw, unprocessed facts. ✓
 - Information is the result of the processing of data and should be useful and meaningful. ✓
- OR
- Any suitable explanation (2)
- 4.1.2 *Describe or Name any THREE of: ✓✓✓*
- Currency/Up to date – This relates to how recent or up to date the data is e.g. daily capturing of sales data.
 - Relevance/useful – This relates to how closely the data matches your needs e.g. need to know when to purchase/how much stock to purchase.
 - Completeness – This refers to how comprehensive the data is e.g. product name and number of items sold must be entered in order to update database.
 - Accurate/correct data.
 - Valid data.
 - Unnecessary information is not included. *Any correct example in each case*
- Do not accept: Normalised data (3)
- 4.2.1 *Any TWO of: ✓✓*
- Manage and maintain databases.
 - Control the allocation and supervision of users of the databases.
 - Allocate user access rights – such as username and passwords.
 - Perform routine maintenance of the database.
 - Making backups.
 - Ensuring security of database.
- Do not accept: design of database (2)
- 4.2.2 The user types in SQL code instead of data ✓ which causes the database to do things the program does not expect it to do. ✓ (2)
- 4.3.1 The database contains two tables/more than one table that have a relationship that is formed ✓ via link between a primary key (SupplierID in SuppliersTb) and a foreign key (SupplierID in OrdersTb).
OR
Indication that implies that the tables are linked together. (1)
- 4.3.2(a) SELECT * FROM OrdersTb ORDER BY QuantityOrdered DESC
- Correct selection and table: SELECT * from OrdersTb ✓
 - Sorting: ORDER BY ✓
 - Correct field in descending order: QuantityOrdered DESC ✓
- Penalise for order of instructions in this question only (3)

- 4.3.2(b) SELECT ItemName, UnitPrice, QuantityOrdered, (UnitPrice * QuantityOrdered) AS [Total Cost] FROM OrdersTb
- Correct fields: ItemName, UnitPrice, QuantityOrdered ✓
 - Calculation: (UnitPrice * QuantityOrdered) ✓
 - New field name: AS [Total Cost] ✓
 - Do not penalise if incorrect brackets were used. (3)
- 4.3.2(c) UPDATE OrdersTb SET UnitPrice = (UnitPrice * 0.05) + UnitPrice
- Correct instruction and table: UPDATE OrdersTb ✓
 - Correct field to change: SET UnitPrice ✓
 - Correct calculation: = (UnitPrice * 0.05) + UnitPrice ✓ OR UnitPrice * 1.05 (3)
- 4.3.2(d) DELETE FROM OrdersTb WHERE DateOrdered < '#01/01/2013#'
- Correct instruction and table: DELETE FROM OrdersTb ✓
 - Correct condition: WHERE DateOrdered < '#01/01/2013#' ✓ (2)
- 4.3.2(e) SELECT Name, ContactNumber, ItemName FROM SuppliersTb, OrdersTb WHERE (YEAR (DeliveryDate) = 2015) AND (SuppliersTb.SupplierID = OrdersTb.SupplierID)
- Correct tables:SuppliersTb, OrdersTb ✓
 - Correct function: YEAR ✓(DeliveryDate) = 2015 ✓
- Join tables: SuppliersTb.SupplierID = OrdersTb.SupplierID ✓ (4)
- OR**
- SELECT Name, ContactNumber, ItemName FROM SuppliersTb INNER JOIN OrdersTb ON SuppliersTb.SupplierID = OrdersTb.SupplierID WHERE YEAR (DeliveryDate) = 2015
- OR**
- DeliveryDate >= '#01/01/2015#' and deliveryDate < '#01/01/2016#'
- OR**
- Use between
- Like "%2015%"

TOTAL SECTION D: 25

SECTION E: SOLUTION DEVELOPMENT**QUESTION 5**5.1.1 Any *THREE* of: ✓✓✓

- There is no heading on the form.
- The button caption does not indicate the function of the button.
- Button is not well positioned/Poor logical flow.
- The label instruction (Name) does not indicate full name or first name or surname only.
- Text fields are too large.
- Text fields do not prompt the user for information so the user is not sure what to enter.
- The label instruction (Complete the form) is unclear.
- Removing the default text.
- Any other practical suggestion.

(3)

- 5.1.2
- Button – On Click button event/mouse event/key event
 - Text Field/Edit Box – Focus on field/On click/On enter/mouse event/key event

OR

Any other valid example.

Concepts:

Component ✓

Event ✓

(2)

5.2 Marks are allocated for:

- Correct order of statements (inside and outside loop) ✓
- Check the first character is alphabetical ✓
- Loop from 2 - 4 and check last 3 characters are digits ✓

(3)

Possible solution:

```

Valid ← True

If Code Size = 4 then
    If First Code character is not letter then
        Valid ← False
    For Index ← 2 to 4
        If Character at index is not digit then
            Valid ← False
Else
    Valid ← False

```

5.3.1 Encapsulation is the concept of hiding the detail of an object. ✓ (1)

5.3.2

Learner
- Name : String
- Age : Integer
+ Learner()/ Constructor
+ SetName(Name : String)

Mark allocation:

- Attributes with appropriate data types Name – String✓, Age – Integer ✓
Constructor without parameters ✓
Check in Java that constructor name is same as class, in Delphi will use constructor create
- SetName with correct parameter and data type ✓
- Correct visibility/public and private/accessibility of attributes and methods ✓

Accept programming code (5)

5.4

Number	Count	NumFactors	Count <= Number ?	Number Modulus Count = 0 ?	Display
4	1	0			
		1			
		1	Yes		
		1		Yes	
	2				
		2	Yes		
		2		Yes	
	3				
		2	Yes		
		2		No	
	4				
		3	Yes		
		3		Yes	
	5				
			No		
					3

- Count column increments from 1 to 5 ✓
- Correct values in NumFactors – starting at 1✓ up to 3 ✓
- Correct values in Count <= Number ✓
- Correct values in Number modulus Count = 0 ✓
- Displaying last value of NumFactors at the end ✓ (6)

NOTE: If a mistake is made in a column but the logic continues to be correct then allocate the marks for the other columns.
2 marks: if only last row written down

5.5 Mark allocation:

- Outer loop with correct range ✓
- Inner loop with correct range ✓
- If statement comparing values in array ✓
- Perform swap of values ✓✓

(5)

Possible solutions where loops begin at 1 for array index.

NOTE: Loops may also begin at zero (0) for the array index.

Possible solution 1 – bubble sort

```
arrMarks ← (67,65,78,83,67,72,65)

For LoopCount1 ← 1 to arrMarks (size - 1)/6/7
  For LoopCount2 ← 1 to arrMarks size-LoopCount1/(size-1)/6
    If arrMarks[LoopCount2] < arrMarks[LoopCount2 + 1] then
      Swap { Temp ← arrMarks[LoopCount2]
            { arrMarks[LoopCount2] ← arrMarks[LoopCount2 + 1]
            { arrMarks[LoopCount2 + 1] ← Temp
```

Possible solution 2 (alternate solution) – selection sort

```
arrMarks ← (67,65,78,83,67,72,65)

For LoopCount1 ← 1 to arrMarks (size - 1)(/6/7)
  For LoopCount2 ← LoopCount1 + 1 to arrMarks size
    If arrMarks[LoopCount1] < arrMarks[LoopCount2] then
      Swap { Temp ← arrMarks[LoopCount1]
            { arrMarks[LoopCount1] ← arrMarks[LoopCount2]
            { arrMarks[LoopCount2] ← Temp
```

TOTAL SECTION E: 25

SECTION F: INTEGRATED SCENARIO**QUESTION 6**

- 6.1.1 *ANY TWO of:* ✓✓
- Faster performance
 - Security more sophisticated
 - Administration is easier
 - Improved control of users and user access to resources
 - Many users
- (2)
- 6.1.2 *ANY TWO of:* ✓✓
- Higher performance RAM/primary memory/larger RAM
 - Higher performance CPU/faster than desktop CPU
 - Higher performance/high speed hard drives/storage/secondary memory
 - Larger HDD
 - Additional/Redundant power supply
- (2)
- 6.1.3(a) *ANY ONE of:* ✓
- Virtual Box
 - Parallels
 - VM Ware
 - Microsoft Hyper-V
 - Any other valid example
 - Do not accept remote desktop.
- (1)
- 6.1.3(b) The physical server runs a host operating system which is independent of the virtual servers that are hosted. ✓ The virtual servers run guest operating systems which are run on top of the current host operating system. ✓
- Concepts:
Base software
Host software
- (2)
- 6.2.1
- Automatically ✓ updates feeds so no reason to visit each site for new news
 - Notifications of changes to content ✓ are passed to user.
- OR
- Saves time as it efficiently scans lots of sites/scans headlines of many articles.
 - Prevents clutter by only showing latest important items.
 - Allows the user to use whichever application they would like to view the content/Read feeds from within the browser/Do not have to switch programs.
- Concepts:
Automatic process
only shows changed content
- (2)
- 6.2.2 *Lossy compression* compresses data by losing some of its quality/poor resolution. ✓
- (1)

- 6.2.3 *ANY TWO of: ✓✓*
- The mobile version allows the page to automatically resize to fit the screen/different layout.
 - The mobile version will make use of smaller images and files/less data.
 - The mobile version is a light weight version of the full website.
 - Different flash player. (2)
- 6.2.4
- Establishes user needs/profile ✓
 - Provides content personalised for specific users/interactive ✓
- Or
- Accept for 1 mark each
- Semantic web that can interpret information like humans.
 - Defined by its mobile capabilities.
 - More compatible in terms of hardware and software. (2)
- 6.3.1 *ANY ONE of: ✓*
- Ensure URL begins with https.
 - Lock sign/secure symbol on the browser. (1)
- 6.3.2 Encryption the scrambling text/data ✓ into a format that cannot easily be read by someone else, using a specified set of rules ✓ /algorithm/key/formula. (2)
- 6.3.3 *ANY TWO of: ✓✓*
- Use a reasonable number of characters (at least 8).
 - Use a mixture of alphabetical and numerical characters.
 - Use both uppercase and lowercase characters.
 - Do not use everyday words like pass/cat etc.
 - Include special characters.
 - Do not include personal information. (2)
- 6.4.1 *ANY TWO of: ✓✓*
- Promotes awareness of the school's activities.
 - Allows feedback from the community/provides an interactive platform.
 - Easy and convenient way to communicate with learners and parents.
 - Sharing of media such as photos and videos.
 - Marketing.
 - Any other valid example. (2)
- 6.4.2 Mark allocation
- ✓ public access to information
 - ✓ negative effect
- Learners' information will be publicly available and could be used for cyber-crimes/stalking/bullying.
- OR
- Social engineering techniques could be used to gain access to other confidential information using the learners' information.
- OR
- Any other acceptable example (2)

- 6.4.3 *ANY TWO of: ✓✓*
- Network managers need to monitor what users do on the network.
 - Clear and visible action must be taken if there is a violation of the AUP.
 - Network restrictions need to be put in place to prevent unacceptable use.
 - AUP needs to be actively communicated to learners.
 - Any other valid example. (2)
- 6.5.1 *ANY TWO of: ✓✓*
- Learners become responsible for their own learning.
 - Material is self-paced.
 - Immediate feedback.
 - Availability of information.
 - More attractive interesting learning.
 - Affordable.
 - Ease of access.
 - Mobile devices can be used in the learning process
 - Any other valid example. (2)
- 6.5.2
- Downloading: the entire file is downloaded before you can start watching. ✓ / One copy is saved and can be reused
 - Streaming: the file is played as it is streamed. ✓/Always need to obtain the file online (2)
- 6.5.3 *ANY TWO of: ✓✓*
- A high speed/high bandwidth Internet connection
 - Permanent/dedicated connection
 - Reliable and stable/strong connection
 - Affordable connection
 - Need a symmetrical connection/High upstream bandwidth (2)
- 6.6.1 *ANY TWO of: ✓✓*
- Receive and send work electronically.
 - Make use of devices for quizzes/assessments/tests/voting.
 - Create media such as videos and podcasts that can be downloaded to mobile devices.
 - Anytime, anywhere/portable access to content.
 - Use mobile technology as a textbook.
 - Any other acceptable answer. (2)
- 6.6.2 *ANY TWO of: ✓✓*
- Set the device to switch the screen off when not in use.
 - Do not run unnecessary apps.
 - Switch off back lighting.
 - Use power saving mode.
 - Turn off connectivity options such as GPS.
 - Turn off WiFi, Bluetooth.
 - Limit the use of music/video/media files (2)

- 6.7.1 *ANY TWO of: ✓✓*
- Internet connection
 - Services as Office 365/Google Docs/DropBox make Online collaboration possible Skype/VoIP client/video conferencing
 - VPN/remote connection
- (2)
- 6.7.2 *Any ONE of ✓*
- No socialisation
 - No physical interaction/No face to face contact
 - Sometimes difficult to assist people over the Internet.
 - Free time could be abused/Need to be online or available all the time.
 - Make excuses to stay away from work/Could take time off from work and say he is ill./Integrity issues
- (1)
- 6.8 *ANY TWO of: ✓✓*
- The school will have access to the software for a monthly fee/
 - Do not have to buy the software and pay for upgrades
 - Software is available on any computer or device anytime, anywhere/easy access to software.
 - Newest version of software.
 - Less hardware requirements.
- (2)

TOTAL SECTION E: 40
GRAND TOTAL: 150