higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

NATIONAL CERTIFICATE (VOCATIONAL)

INTRODUCTION TO SYSTEMS DEVELOPMENT

(Second paper)

NQF LEVEL 2

(10041022)

13 March 2020 (Y-paper)
13:00–17:00

This question paper consists of 19 pages.

565Q2S2013

Copyright reserved

TIME: 4 HOURS
MARKS: 100

INSTRUCTIONS AND INFORMATION

1. Answer all the questions.
2. Read all the questions carefully.
3. Number the answers according to the numbering system used in this question paper.

INSTRUCTIONS TO STUDENTS AND LECTURERS

1. Students must be seated in the computer laboratory 30 minutes before the start of the examination to ensure that all computers have the appropriate programs.
2. These questions must be completed on a computer by using MS Office 2003 or a later version and VB.NET 2005 or VB.NET Express or VB.NET 2008.
3. Students must ensure that they print their work immediately after they have completed a question. Ensure that your EXAMINATION NUMBER is on all printouts. No questions without a printed EXAMINATION NUMBER will be marked.
4. In the event of problems such as a power failure, computer breakdown or a printer breakdown, the invigilator will make the necessary arrangements for the student(s) to continue with the examination process once the problem has been resolved. Students will not be penalised for time lost under these circumstances.
5. All work on the computer should be saved at regular intervals to prevent loss of work. No additional time may be allowed for work lost due to lack of saving.
6. NO student may print his/her work for another student or make a memory stick available to another student or access another student's work on the network. Any attempt to access any information from or transfer information to another student in whatever manner is a contravention of the examination rules and regulations and will be viewed in an extremely serious light.

Copyright reserved

7. All PRINTOUTS and screen prints done during the session must be handed in.

8. Ensure that your EXAMINATION NUMBER appears on all printouts.

9. Students may not copy the source code from VB.NET to MS Word and then print it from the word processor.

10. Steps to print a VB.NET form or any other screen:

    10.1 Run the solution.

    10.2 Hold the ALT key down and press the print screen button. (This will copy the active screen.)

    10.3 Open MS Word and paste the form/screen in a new document.

    10.4 Add your EXAMINATION NUMBER to the top right-hand side of your page.

    10.5 Print the page.

11. You are allowed to use the help files in VB.NET.

12. The following files appear on your hard drive in the folders:

    12.1 QUESTION 1
    … \TVGuide

    IMAGE: tvguide.jpg

    … \IMAGES

    12.2 QUESTION 2
    … \MultiplicationTable

    12.3 QUESTION 3
    … \BondCalculator

    12.4 QUESTION 4
    … \FastFood

Copyright reserved

QUESTION 1

The following questions refer to the VB.NET program, ***TVGuide***, that has been saved on your computer.

You are hired by a new up-and-coming satellite television company to design and create the user interface for their new electronic TV guide.

The user first selects the package he/she prescribes to and then selects the categories he/she would like to include. Thereafter, the user selects the show times (morning, afternoon or evening) and the date and then clicks the *Display Guide* button.

All the channels, shows and times for that time of day are then displayed in a list box.

NOTE: No code is required for this question; you are only required to create the graphical user interface.

Use FIGURE 1 below and the instructions that follow as a guide to help you complete the interface.

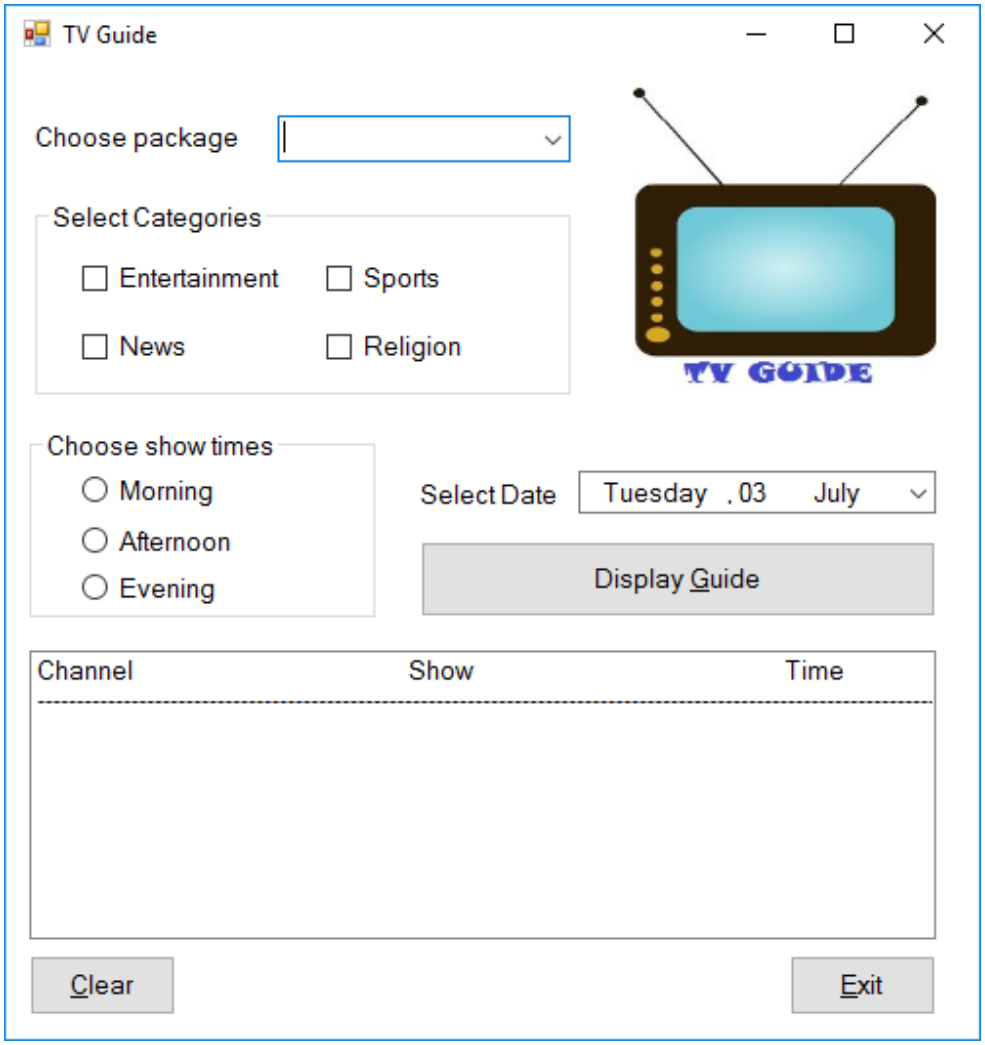


FIGURE 1

Copyright reserved

A blank interface has been saved on your computer. Open the program, ***TVGuide***, and follow the instructions below to complete the interface.

1.1 Add a PICTURE BOX to the form as part of the heading:

CONTROL	NAME	IMAGE
PictureBox	picTV	Use the image saved on your computer as 'tvguide.jpg'

(2)

1.2 Add TWO GROUP BOXES to the form to group the Categories and Show Times:

CONTROL	NAME	TEXT
Group Box	grpCategories	Select Categories
Group Box	grpShowTimes	Choose show times

(2)

1.3 Add TWO LABELS to the form – ONE for the package and ONE for the date:

CONTROL	NAME	TEXT	PROPERTIES
Label	lblPackage	Choose Package	Default
Label	lblDate	Choose Date	Default

(2)

1.4 Add a COMBO BOX to the form to select the package:

CONTROL	NAME
Combo Box	cboPackage

(1)

1.5 Add FOUR CHECK BOXES to the *Categories* group box on the form to allow the user to choose categories:

CONTROL	NAME	TEXT
Check Box	chkEntertainment	Entertainment
Check Box	chkNews	News
Check Box	chkSports	Sports
Check Box	chkReligion	Religion

(4)

1.6 Add THREE RADIO BUTTONS to the *Show Time* group box on the form to indicate the show times:

CONTROL	NAME	TEXT
Radio Button	radMorning	Morning
Radio Button	radAfternoon	Afternoon
Radio Button	radEvening	Evening

(3)

Copyright reserved

1.7 Add a DATETIMEPICKER to the form to select the date:

CONTROL	NAME	DEFAULT DATE
DateTimePicker	dtpDate	Today's date

(2)

1.8 Add a LIST BOX to the form to display the channels, shows and times:

CONTROL	NAME	Items
List Box	lstDisplay	Channel Show Time ----------

(4)

SEE FIGURE 1

1.9 Add THREE BUTTONS to the form – ONE to display show details, ONE to clear the controls on the form, and ONE to exit the program:

CONTROL	NAME	TEXT	Font Size
Button	btnDisplay	Display &Guide	Default
Button	btnClear	&Clear	Default
Button	btnExit	&Exit	Default

(6)

1.10 Change the text in the title of the form to your EXAMINATION NUMBER and make a printout of the form. (2)

[28]

QUESTION 2

The following questions refer to the partially completed VB.NET program, ***MultiplicationTable*** that has been saved on your computer.

As a mathematics student, you constantly use your calculator in order to do simple multiplication problems. You decide to write a program that will automatically generate the multiplication table up to ten for any number in order to not rely on your calculator as much.

The program takes as input any whole number and then calculates and displays the multiplication table up to 10 for that number i.e. 1xn, 2xn,….,10xn; where n is the number inputted.

FIGURE 2 on the next page is a screenshot of what the user interface looks like. The completed interface with partially completed code has been saved on your computer. Open the program, ***MulitplicationTable***, and follow the instructions to complete the program:

Copyright reserved

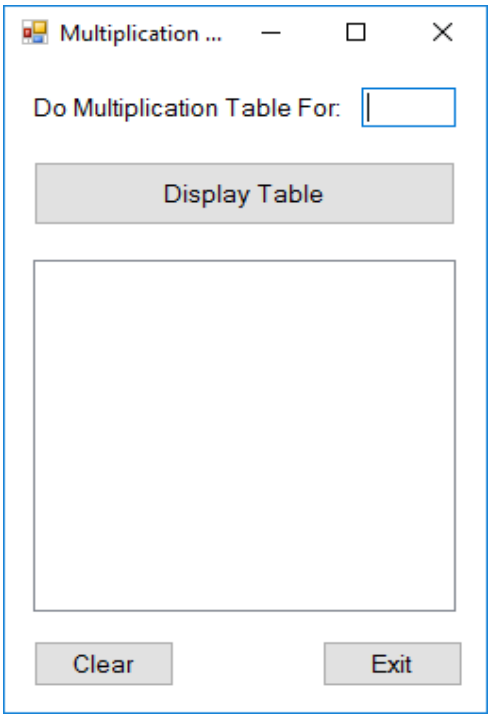


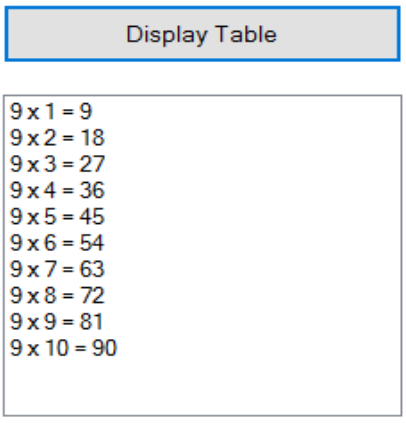
FIGURE 2

The following steps explain how to use the program in order to calculate and display the multiplication table up to 10 for any number:

STEP 1: RUN the program and enter the number for which you want the multiplication table:



STEP 2: Click the *Display Table* button to calculate and display the multiplication table up to 10:



Copyright reserved

STEP 3: The *Clear* button clears the list box and the textbox, while the *Exit* button closes the program.

2.1 Add the following program code to the click event of the *Display Table* button:

2.1.1 Declare the outstanding required variables. (2)

2.1.2 Convert and assign the value from the *Number* text box to the appropriate variable. (1)

2.1.3 Complete the 'to' section of the *For-Next* loop. (2)

2.1.4 Complete the calculation in the body section of the *For-Next* loop. (2)

2.1.5 Add the appropriate item to the list box for each iteration of the loop. ***See Step 2 above.*** (4)

2.2 Add the following program code to the click event of the *Clear* button:

2.2.1 Clear the *number* textbox.

2.2.2 Clear the list box.

2.2.3 Move the focus to the *number* textbox

(3 × 1) (3)

2.3 Clicking the *Exit* button should close the program. (1)

Make a PRINTOUT of the source code.

[15]

QUESTION 3

The following questions refer to the VB.NET program, ***BondCalculator***, that has been saved on your computer.

A new startup home loan company is trying to make waves in the market with their new innovative way of calculating the interest prospective home buyers would have to pay if they decided to take up a home loan with them.

They hire you, a software developer, to create a program that will show how much interest their customers would pay based on a specific *loan amount, interest rate* and *loan term.* The customer can then also choose from some optional extra services that the company provides.

The program should calculate and display all of the following:

- Basic Monthly Installment
- Monthly Interest
- Total Monthly Repayment
- Lifetime Interest Payment (Total interest over the loan period)
- Total Bond Repayment (Interest + initial loan amount)
- Extra Services

Use the IPO chart below to help you with the program:

INPUT	PROCESSING	OUTPUT
Loan Amount Interest Rate Select Term Select Extra Services	**Terms** • 10 years–120 (months) • 15 years–180 (months) • 20 years–240 (months) **Extra Services:** • Bond Protection – R15000 • Legal Aid – R25000 • Bond Initiation – R12000 **BasicMonthlyInstalment** = LoanAmount / Term **MonthlyInterest** = LoanAmount * (Interest / 100) / 50 **TotalMonthlyRepayment** = BasicMonthlyInstalment + MonthlyInterest **LIfetimeInterestPayment** = MonthlyInterest * Term **TotalBondRepayment** = LoanAmount + LifetimeInterestPayment	BasicMonthlyInstalment MonthlyInterest TotalMonthlyRepayment LifetimeInterestPayment TotalBondRepayment ExtraServicesTotal

Copyright reserved

FIGURE 3 below is a screenshot of what the user interface looks like. The completed interface has been saved on your computer. Open the solution, ***BondCalculator***, and follow the instructions to complete the program:

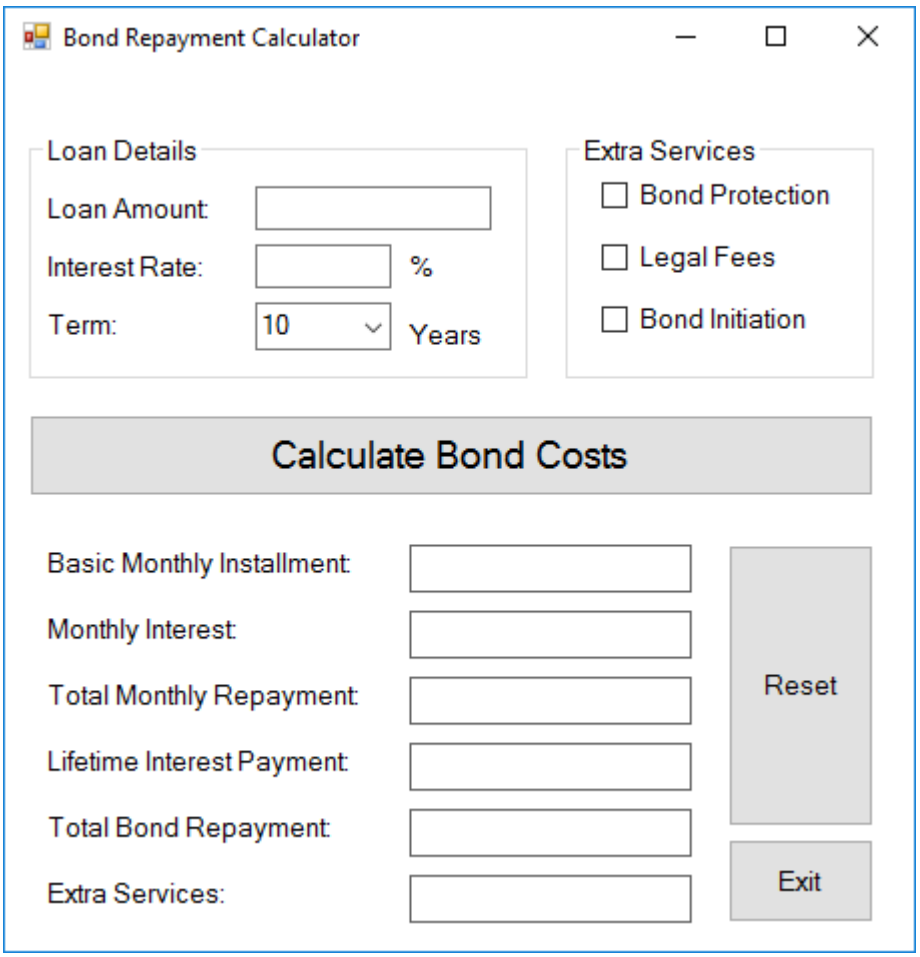
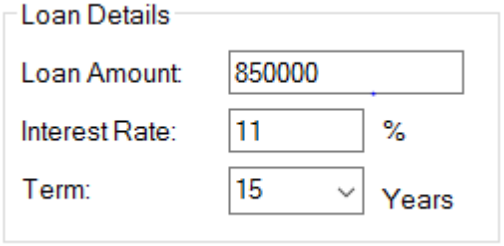


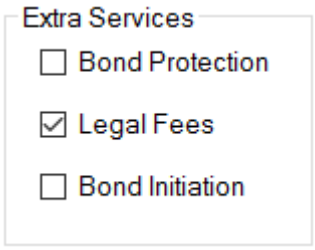
FIGURE 3

The following steps explain how to use the program in order to calculate the required values:

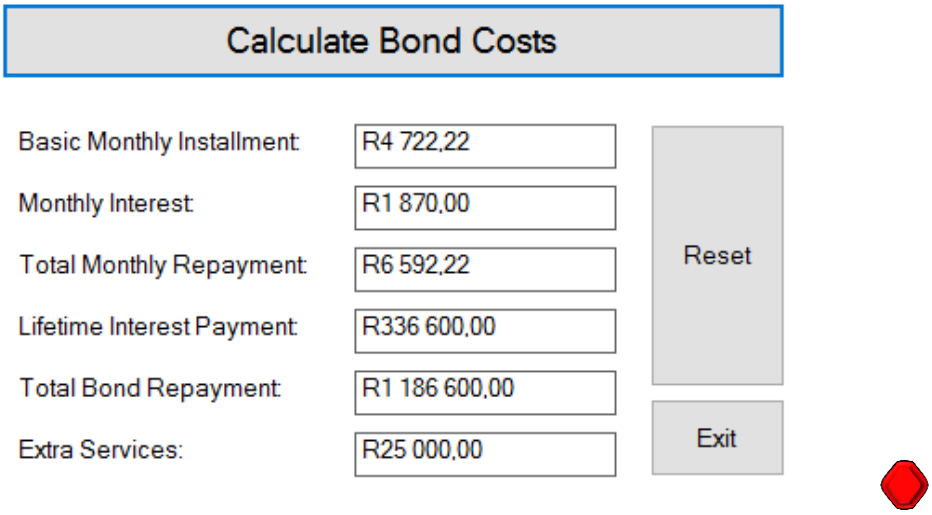
STEP 1: RUN the program and enter the *loan amount* and *interest rate,* and select the *term*:



STEP 2: Select the *extra services:*



STEP 3: Click the *Calculate Bond Costs* button to calculate and display ALL the required values:



STEP 4: The *Reset* button clears the textboxes, check boxes and display labels, and also selects the first item in the combo box. The *Exit* button closes the program.

3.1 Add the following code to the *Form_Load* event:

3.1.1 Add the following THREE items to the *Term* combo box:

- 10
- 15
- 20 (1)

3.1.2 Select the first item in the *Term* combo box. (1)

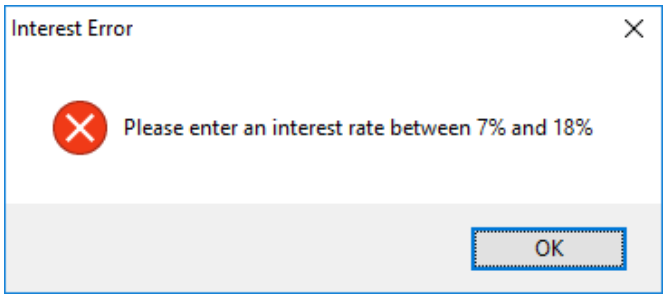
3.2 Add the following program code to the click event of the *Calculate Bond Costs* button:

3.2.1 Declare all the required variables. (4)

3.2.2 Convert and assign the values from the *Loan Amount* and *Interest Rate* text boxes to the appropriate variables. (2)

Copyright reserved

3.2.3 If the *Interest Rate* is not between 7 and 18, the following message should be displayed, and the program should exit the sub procedure. The focus should then be moved to the *Interest Rate* textbox.



(4)

3.2.4 Determine the *Term* based on the selection in the combo box by using a **<select … case>** statement. (4)

3.2.5 Determine the total for the *Extra Services* based on the check box selection(s). (3)

3.2.6 Calculate and display ALL of the following as shown in *Step 3* on page 11:

- Basic Monthly Instalment
- Monthly Interest
- Total Monthly Repayment
- Lifetime Interest Payment
- Total Bond Repayment (5 × 3) (15)

3.2.7 Display the amount for the *Extra Services* as shown in *Step 3* on page 11. (1)

3.3 Clicking the *Clear* button should clear all the text boxes, check boxes and display labels, as well as select the first item in the combo box. (4)

3.4 Clicking the *Exit* button should close the program. (1)

Make a PRINTOUT of the source code.

[40]

QUESTION 4

The following questions refer to the VB.NET program, ***FastFood*** that has been saved on your computer.

You are hired buy a local fast food restaurant to fix up a program that was partially completed by a different software company that they hired before you. The company created the program but then could not figure out why it was not working the way it was supposed to, and consequently gave up on trying to get it to work.

The restaurant informs you that they are happy with the user interface and that they just wanted the required functionality to work the way it was meant to.

FIGURE 4 below is a screenshot of what the user interface looks like:

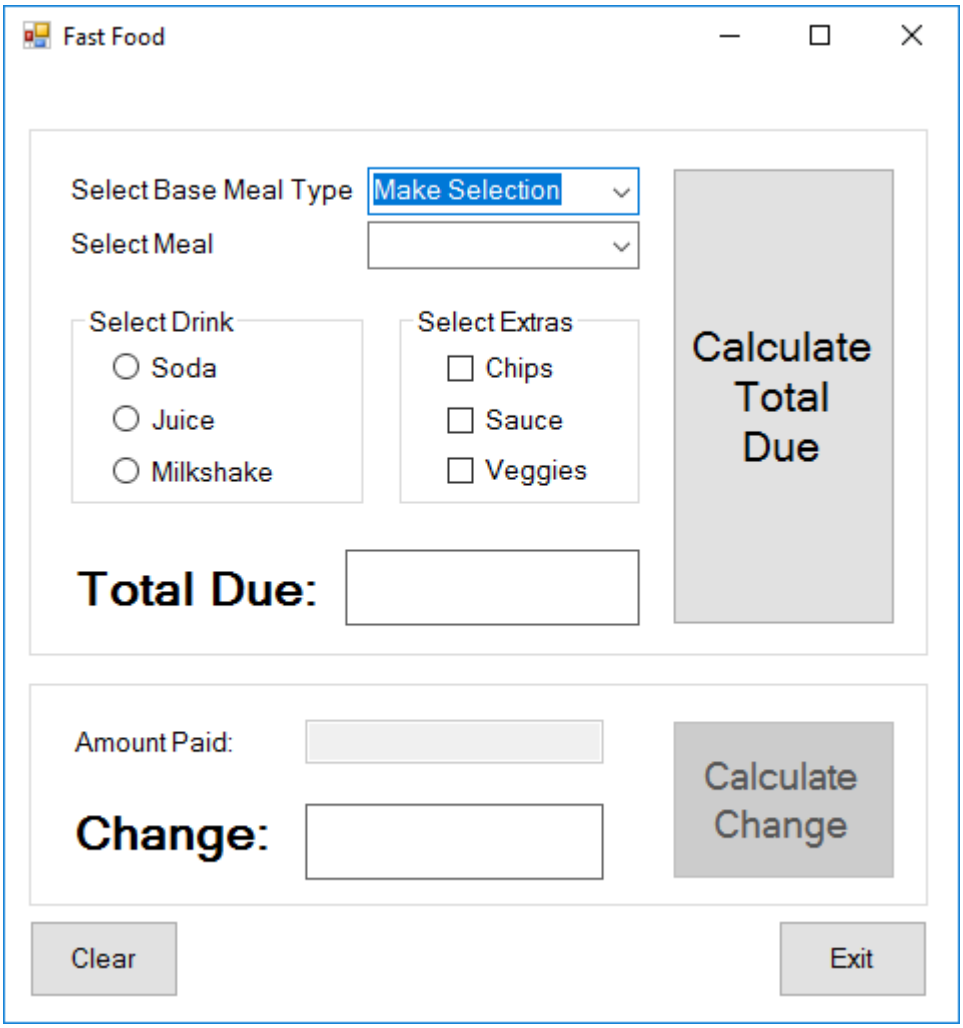


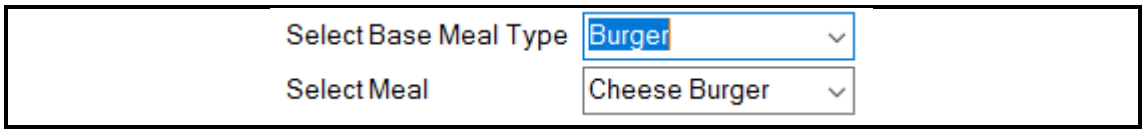
FIGURE 4

Copyright reserved

The following steps explain how the program was supposed to work and what the restaurant would like you to implement using the partially completed code:

STEP 1: RUN the program and select the base meal type. After making the selection, the *Meal* combo box is automatically populated based on the base meal type selection. The following are the base meal types with their respective meals:

- Burger
  - Cheese Burger
  - Chicken Burger
- Steak
  - Sirloin
  - Rump
- Pasta
  - Ravioli
  - Lasagne



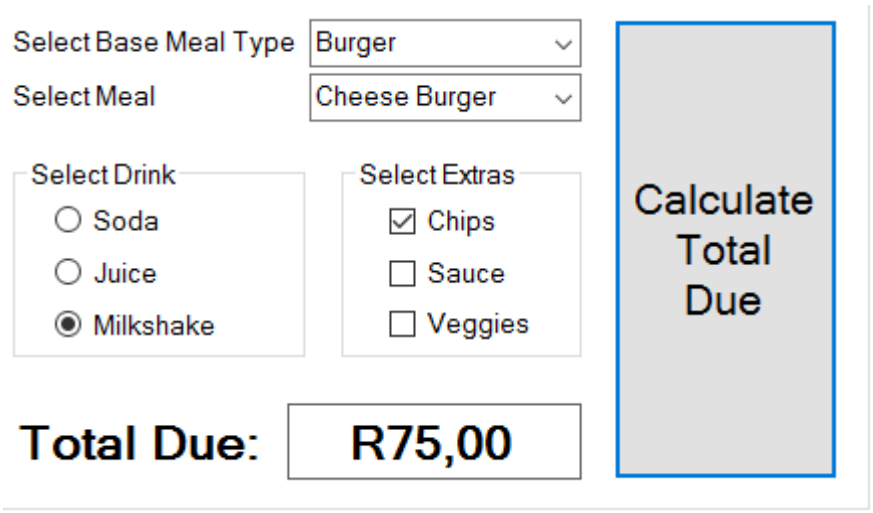
STEP 2: Select the drink type and extras, which are priced as follows:

DRINKS	EXTRAS
• Soda: R17 • Juice: R22 • Milkshake: R35	• Chips: R15 • Sauce: R10 • Veggies: R18

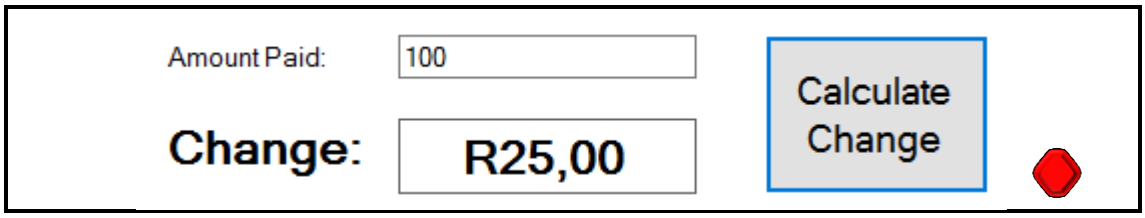


Copyright reserved

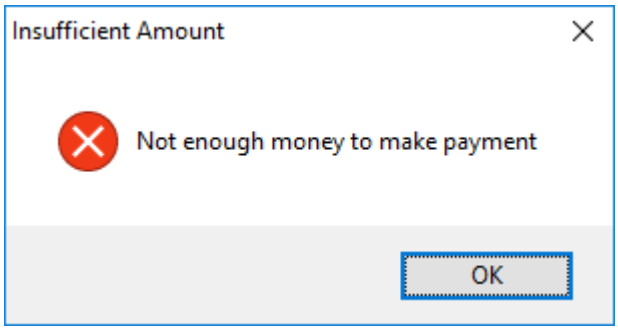
STEP 3: Click the *Calculate Total Due* button to display the total due. This also enables the *Calculate Change* button:



STEP 4: Enter the amount paid and click the *Calculate Change* button to display the change.



STEP 5: If the customer did NOT pay enough, the following message should be displayed:



STEP 6: Clicking the *Clear* button does the following:

- Clears the textbox.
- Clears radio buttons and check boxes.
- Clears the display labels.
- Selects the first item in the *Base Meal* combo box.
- Clears the *Meal* combo box.
- Disables the *Calculate Change* button.

Copyright reserved

STEP 7: Clicking the *Exit* button closes the program.

You look at the code that the other company created and realise that it contains numerous syntax and logical errors, and you now have to debug the source code.



Below is a printout of the program code with all the syntax and logical errors still present:

```
Option Strict On
Option Explicit On

Public Class Form1
    'Examination Number Here
    decTotalDue As Decimal = 0

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Add Items to combo box
        cboBaseMeal.Items.Add("Make Selection")
        cboBaseMeal.Items.Add("Burger")
        cboBaseMeal.Items.Add("Steak")
        cboBaseMeal.Items.Add("Pasta")
        cboBaseMeal.SelectedIndex = 0
End Sub

Private Sub cboBaseMeal_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboBaseMeal.SelectedIndexChanged
'Add items to Meal combo box based on selection in base meal combobox
        Select Case cboBaseMeal.SelectedIndex
            Case 0
                cboMeal.Clear()
            Case 1
                cboMeal.Items.Clear()
                cboMeal.Items.Add("Sirloin")
                cboMeal.Items.Add("Rump")
                cboMeal.SelectedIndex = 0
            Case 2
                cboMeal.Items.Clear()
                cboMeal.Items.Add("Cheese Burger")
                cboMeal.Items.Add("Chicken Burger")
                cboMeal.SelectedIndex = 0
            Case 3
                cboMeal.Items.Clear()
                cboMeal.Items.Add("Ravioli")
                cboMeal.Items.Add("Lasagne")
                cboMeal.SelectedIndex = 0
End Sub
```



Copyright reserved

```
Private Sub btnTotalDue_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnTotalDue.Click
        Dim decMealCost As Decimal = 0
        Dim decDrinkCost As Decimal = 0
        Dim decExtrasCost As Decimal = 0

        'Determine Meal Cost
        Select Case cboBaseMeal.SelectedIndex
            Case 0
                MessageBox.Show("Please make a base selection")
                Exit Sub
            Case 1
                If cboMeal.SelectedIndex = 0 Then
                    decMealCost = 25
                ElseIf cboMeal.SelectedIndex = 1 Then
                    decMealCost = 29
                End If
            Case 2
                If cboMeal.SelectedIndex = 0 Then
                    decMealCost = 79
                ElseIf cboMeal.SelectedIndex = 1 Then
                    decMealCost = 69
                End If
            Case 3
                If cboMeal.SelectedIndex = 1 Then
                    decMealCost = 35
                ElseIf cboMeal.SelectedIndex = 0 Then
                    decMealCost = 44
                End If
        End Select

        'Determine drink cost
        If radSoda.Checked = True Then
            decDrinkCost = 17
        ElseIf radJuice.Checked = False Then
            decDrinkCost = 22
        ElseIf radShake.Checked = True Then
            decDrinkCost = 35
        End If
        'Determine Extras cost
        If chkChips.Checked = True Then
            decExtrasCost = 15
        End If
        If chkSauce.Checked = True Then
            decExtrasCost = 10
        End If
        If chkVeggies.Checked = True Then
            decExtrasCost = decExtrasCost + 18
        End If
        'Calculate Total Due
        decTotalDue = decMealCost + decDrinkCost + decExtrasCost



        lblTotalDue.Text = decTotalDue.ToString("C2")

        'Enable Amount Paid textbox and Change button
        txtAmountPaid.Disabled = False
        btnChange.Enabled = True

End Sub
```

Copyright reserved

```
Private Sub btnChange_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnChange.Click
        Dim decAmountPaid As Decimal = 0
        Dim decChange As Decimal = 0

        decAmountPaid = CDec(txtAmountPaid.Text)
        'Determine if amount paid was enough
        If decAmountPaid > decTotalDue Then
            MessageBox.Show("Not enough money to make payment",
"Insufficient Amount", MessageBoxButtons.OK, MessageBoxIcon.Error)
            txtAmountPaid.Clear()
            txtAmountPaid.Focus()
            Exit Sub
        End If
        'Calculate and display change
        decChange = decAmountPaid - decTotalDue
        lblChange = decChange.ToString("C2")
End Sub

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
        'Clear controls
        cboBaseMeal.SelectedIndex = 0
        cboMeal.Items.Clear()
        cboMeal.Text = ""
        radSoda.Checked = False
        radJuice.Checked = False
        radShake.Checked = False
        chkChips.Checked = False
        chkSauce.Checked = False
        chkVeggies.Checked = False
        lblTotalDue.Text = ""
        txtAmountPaid.Clear()
        lblChange.Text = ""
        txtAmountPaid.Enabled = False
        btnChange.Enabled = False
End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        'exit program
        Close()
End Sub.
```

4.1 The completed program has been saved on your computer, but the program is not functioning properly.

Open the program, ***FastFood***, and correct all the syntax and logical errors. (14)

4.2 Add your EXAMINATION NUMBER to the source code as indicated and make a PRINTOUT of the source code. (1)

4.3 Add you EXAMINATION NUMBER to the title bar of the form, and then use the program to calculate the total due and change given for the following input:



Base Meal	Steak
Meal	Rump
Drink	Milkshake
Extras	Chips and Sauce
Amount Paid	R150

Make a PRINTOUT of the form showing the correct result. (2)
[17]

TOTAL: 100

Copyright reserved