

Second Assignment – Problem Solving Task

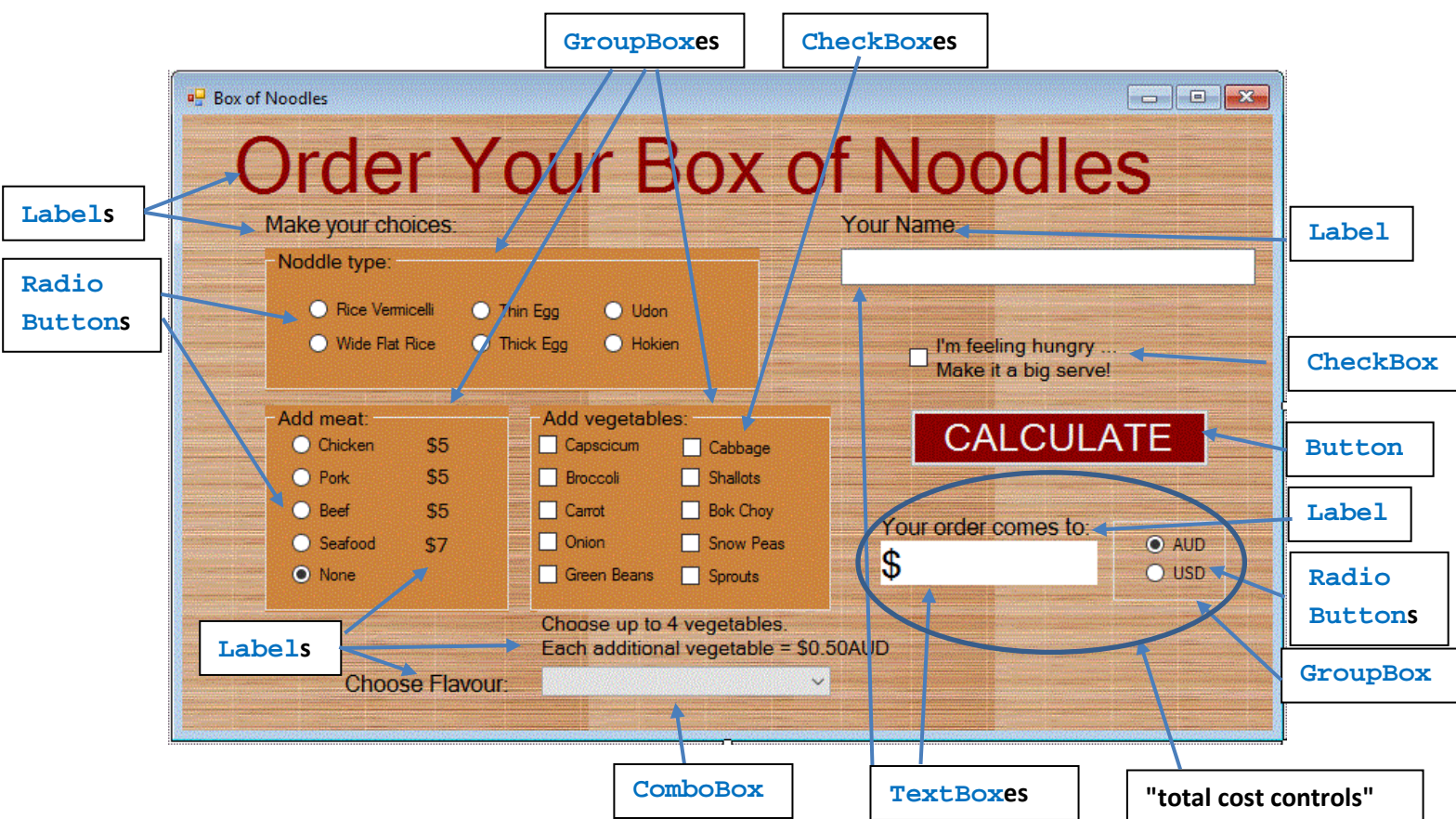
Box of Noodles

Weighting 30%

Specification

You are to implement a simple online ordering form for a fictitious noodle restaurant called "Box of Noodles" - see screenshot in Figure 1. You are required to use **Windows Forms**, rather than other GUI technologies such as WPF, XNA, web pages etc. Also, to keep this program relatively simple, do not use advanced techniques such as MVC or layered architectures. (If you do not know what these are, that's fine, just ignore them.)

You are required to develop a GUI layout with the same widgets as shown in the screenshot in Figure 1. You are not expected however to replicate the exact location, size, colour etc. of each control. Choose your own background. The look and feel of the widgets are your own choice. However, the functionality provided by your GUI must meet the specifications contained in this document. The GUI must be be user-friendly and intuitive.



The GUI will consist of 1 **Button**, 11 **CheckBoxes**, 1 **ComboBox**, 4 **GroupBoxes**, 10 **Labels**, 13 **RadioButtons** and 2 **TextBoxes**. Not all controls will be visible or enabled at the start of the program.

Motivation

Many take-away restaurants provide an option for on-line ordering, which is both convenient for customers as well as potentially cost efficient for businesses. Your program is to provide some of the functionality that might be required for on-line ordering of noodle meals. It will by no means replicate a complete on-line ordering system as this assignment focuses on the 'front end' of such a system (GUI), with minimal 'back end' functionality.

Program Functionality

The GUI program you are to write will provide the options for the user to choose from a limited number of ingredients, as well as an option to upsize to a larger serve, and pay in either Australian or US dollars.

User Input

The user will need to construct the noodle dish by choosing:

- one (1) type of noodle
- one (1) type of meat (or none at all)
- any number of vegetables (with an additional charge for more than 4 vegetables)

Noodles

The user must choose one of the available noodle types. As it doesn't make sense to order a box of noodles without noodles, if the user fails to select a noodle type, the program must display a **MessageBox** with an appropriate message before any calculation is made. All noodle types are charged at a cost of \$5.50AUD.

Meat

The user can choose any of the meats available, or "None" if they do not want meat or seafood in their meal. Chicken, pork and beef are charged at \$5.00AUD and seafood at \$7.00AUD.

Vegetables

The user can choose any number of vegetables to add to their meal. The first four (4) are included in the price of the meal. Every additional vegetable will be charged at \$0.50AUD (50 cents). It is possible to have a noodle meal without any vegetables (although not as tasty!), so the program should allow for this.

Flavour

The user must choose a flavour (sauce base) for their noodles. The available flavours are:

VERY Hot and spicy
MEDIUM Hot and spicy
MILD spicy
Sweet chilli
Soy and honey

As a noodle meal would be very dry and uninteresting without any sauce, if the user fails to select a flavour from the list, the program must display a **MessageBox** with an appropriate message to select a sauce before any calculation can be made.

Up-sizing

The customer has the option to up-size their meal, in which case the cost of their meal will increase by 50%.

Customer Name

In order to identify the customer, they must supply their name. If the user fails to supply a name, the program must display a message box with an appropriate message to supply a name before the calculation can be made.

Output

When the necessary information has been entered and the 'calculate' **Button** hit by the user the following calculations are made:

- The total of the order includes charges for:
 - noodles
 - meat (if any)
 - vegetables (if more than 4)
 - upsizing charge (if any)

The order total will be displayed in the total cost **TextBox** in currency format, initially in AUD (Australian dollars). The user has the option to change the display to USD (US dollars).

GUI Functionality

You may design your GUI however you prefer, as long as you have all of the same widgets as described above and as shown in Figure 1.

GUI at start of execution

Figure 2 shows the example GUI when the program first starts. Note that:

- none of the noodle type **RadioButtons** are selected.
- in the meat section, the **RadioButton** for "None" is selected.
- none of the vegetable option **CheckBoxes** are selected.
- no flavour option is displayed.

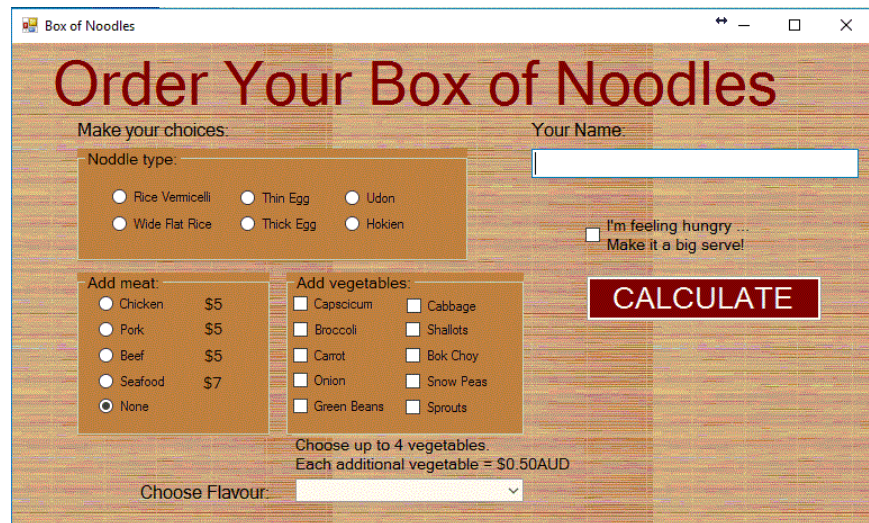


Figure 2: GUI at Startup

- the **TextBox** for the user's name is empty.
- the up-sizing **CheckBox** is unselected.
- the **Button** is enabled and visible.
- the **Label**, **TextBox** and **RadioButtons** for displaying the total cost (the "total cost controls") are not visible.
- all other controls are as per the screenshot in Figure 1.

During/after interaction

- No calculation should be made without:
 - a noodle type selected;
 - a flavour chosen; and
 - the customer name entered.

In the case of any of these missing, a **MessageBox** should display an appropriate message informing the user what is missing and required.

- The user must choose from the supplied list of flavours, and must not be able to enter another flavour.
- Once all the required selections and input have been made, hitting the **Button** will cause the cost of the meal to be calculated. The "total cost controls" (see Figure 1) will be visible, with the calculated cost displayed in the **TextBox**. Being a currency, the "\$" must precede the amount which must be displayed to two (2) decimal places. The currency selected by default will be AUD.
- The user must not be able to edit the total amount displayed.
- The user will be able to change the currency from/to AUD or USD, with the new amount displayed in the **TextBox**.

- Changes to any selection of meat or vegetable will cause the "total cost controls" to be invisible, thereby requiring a recalculation of the total cost. When the calculate **Button** is hit again, the total cost is calculated again and the value displayed in the currency that had previously been selected.
- Changes to the up-sizing option (**CheckBox**) will cause the total to be updated automatically. That is, the "total cost controls" remain visible, and the calculate **Button** does not need to be hit.

General Requirements

Though we stress that it is a good programming principle to place the GUI interactions and the program logic into separate classes, for this assignment all methods will be in the `Form1` class.

You must, however, place any program logic into a separate method which is called from event handlers or other methods in your program.

Many of the event handlers will be the same for multiple controls in a group, for example, changing the meat or vegetables **RadioButtons** or **CheckBoxes** after a calculation has been made will cause the total cost controls to disappear. In cases like this, rather than having a separate event handler for each of the controls, there must be a single event handler that each of the controls share. (See Lecture 5 re shared event-handlers.)

Code Quality

The assignment problem is straightforward. The solution of the problem will be a C# project consisting of a single class.

Ensure that your code conforms to the C# Coding Style Guide.

Though the code could be written as *straight line code*, the sheer number of lines would make the solution difficult to read, hard to understand, hard to modify and maintain, and much more likely to contain logical errors. **A straight line code solution will not receive full marks even if it fulfils the required functionality.**

Any identifier (variable or method name) should be **meaningful** and provide a clear indication to the reader of its intended purpose. Method names should be *verbs* ("doing" words). If you find it difficult to meaningfully name the action performed by your method, it is likely that the method has more than one purpose, and should therefore be broken into smaller, single-purpose, methods. There is no upper limit on the number of methods used.

Documentation

The class must be preceded by an XML comment containing your full name, student id and a short overall description of the program. It should be placed between the namespace and class statements.

Each method must be preceded by an XML comment describing its purpose.

Any non-trivial code should also be commented – but only where it adds value to the readability of your solution.

Learning Goals

Goals of the assignment are for you to experience:

- Top-down design & development
- Incremental Development & Incremental Implementation
- Translating simple algorithms into C# code
- Writing user defined methods with/without parameters
- Using procedural programming constructs (sequence, selection)
- Creating a professional looking Graphical User Interface (GUI) which is intuitive by design
- Creating a professional, fully functioning GUI program
- Becoming confident and comfortable with programming in the small.

Academic Integrity

This assignment is for ***individual*** assessment only. That means the work you submit must be your own work and not the work of anyone else. You are not permitted to collaborate with your peers on this assignment, other than to discuss high-level strategies. You are not permitted to show your code to (or allow your code to be seen by) another student, or to look at another student's code. You are not permitted to ask or commission someone else to write the assignment for you (including on-line sources), or help you write the assignment. If you are struggling to complete the assignment yourself, please see one of the teaching staff in consultation as soon as possible, or talk to them in class.

If you are in any doubt as to what is permitted and what is considered a breach of academic integrity, please ask one of the teaching staff.

Final Comments

You should ensure that your project always compiles and does *something*.

This assignment is not so much about fancy GUI design. Instead, aim for a GUI which is easy to use, intuitive and performs the required functions accurately.

Happy noodling!

Appendix - Sample screen shots of interaction with GUI (in order):
(Note that a description appears underneath each screenshot.)

The screenshot shows the 'Box of Noodles' application window. The title bar says 'Box of Noodles'. The main heading is 'Order Your Box of Noodles'. Under 'Make your choices:', there are two sections: 'Noodle type:' with radio buttons for Rice Vermicelli, Thin Egg, Udon, Wide Flat Rice, Thick Egg, and Hokien; and 'Add meat:' with radio buttons for Chicken (\$5), Pork (\$5), Beef (\$5), Seafood (\$7), and None (selected). To the right, 'Your Name:' has a text box containing 'Customer Chris'. Below that is a checkbox 'I'm feeling hungry ... Make it a big serve!'. In the center, 'Add vegetables:' has checkboxes for Capsicum, Broccoli, Carrot, Onion, Green Beans, Cabbage, Shallots, Bok Choy, Snow Peas, and Sprouts. Below this is the text 'Choose up to 4 vegetables. Each additional vegetable = \$0.50AUD'. At the bottom, 'Choose Flavour:' has a dropdown menu. A large red 'CALCULATE' button is on the right. A small dialog box is open in the bottom right corner with the text 'Please select a noodle type!' and an 'OK' button.

No noodle type, meat, vegetables or flavour selected. Name added. Calculate **Button** hit.

The screenshot shows the 'Box of Noodles' application window after some changes. The 'Noodle type:' section now has 'Rice Vermicelli' selected. The 'Your Name:' text box is now empty. The 'CALCULATE' button is still present. The dialog box now says 'Please supply a name for this order!' with an 'OK' button.

Noodle type selected. Name deleted. Calculate **Button** hit.

Name added. Calculate **Button** hit.

The screenshot shows a web browser window titled "Box of Noodles". The main heading is "Order Your Box of Noodles". Below this, there are several sections for customization:

- Make your choices:**
 - Noodle type:** Radio buttons for Rice Vermicelli (selected), Thin Egg, Udon, Wide Flat Rice, Thick Egg, and Hokien.
 - Add meat:** Radio buttons for Chicken (\$5), Pork (\$5), Beef (\$5), Seafood (\$7), and None (selected).
 - Add vegetables:** Checkboxes for Capsicum, Cabbage, Broccoli, Shallots, Carrot, Bok Choy, Onion, Snow Peas, Green Beans, and Sprouts.
- Your Name:** A text input field containing "Customer Chris".
- I'm feeling hungry ...** A checkbox with the text "Make it a big serve!" below it.
- CALCULATE** A large red button.
- Price:** A yellow box displaying "\$5.50".
- Currency:** Radio buttons for AUD (selected) and USD.
- Choose up to 4 vegetables.** Text indicating "Each additional vegetable = \$0.50AUD".
- Choose Flavour:** A dropdown menu showing "VERY Hot and spicy".

Flavour selected. Calculate **Button** hit.

Box of Noodles

Order Your Box of Noodles

Make your choices:

Noodle type:

☒ Rice Vermicelli ☐ Thin Egg ☐ Udon
☐ Wide Flat Rice ☐ Thick Egg ☐ Hokien

Add meat:

☐ Chicken \$5
☐ Pork \$5
☐ Beef \$5
☐ Seafood \$7
☒ None

Add vegetables:

☒ Capsicum ☐ Cabbage
☐ Broccoli ☐ Shallots
☐ Carrot ☐ Bok Choy
☐ Onion ☐ Snow Peas
☐ Green Beans ☐ Sprouts

Choose up to 4 vegetables.
Each additional vegetable = \$0.50AUD

Choose Flavour: VERY Hot and spicy

Your Name: Customer Chris

☐ I'm feeling hungry ...
Make it a big serve!

CALCULATE

One vegetable added. Total cost controls disappear.

Box of Noodles

Order Your Box of Noodles

Make your choices:

Noodle type:

☒ Rice Vermicelli ☐ Thin Egg ☐ Udon
☐ Wide Flat Rice ☐ Thick Egg ☐ Hokien

Add meat:

☐ Chicken \$5
☐ Pork \$5
☐ Beef \$5
☐ Seafood \$7
☒ None

Add vegetables:

☒ Capsicum ☐ Cabbage
☐ Broccoli ☒ Shallots
☐ Carrot ☒ Bok Choy
☒ Onion ☐ Snow Peas
☐ Green Beans ☐ Sprouts

Choose up to 4 vegetables.
Each additional vegetable = \$0.50AUD

Choose Flavour: VERY Hot and spicy

Your Name: Customer Chris

☐ I'm feeling hungry ...
Make it a big serve!

CALCULATE

\$5.50

☒ AUD
☐ USD

Added three more vegetables and calculate **Button** hit. No charge for 4 vegetables

Box of Noodles

Order Your Box of Noodles

Make your choices:

Noodle type:

☒ Rice Vermicelli ☐ Thin Egg ☐ Udon
☐ Wide Flat Rice ☐ Thick Egg ☐ Hokien

Add meat:

☐ Chicken \$5
☐ Pork \$5
☐ Beef \$5
☐ Seafood \$7
☒ None

Add vegetables:

☒ Capsicum ☐ Cabbage
☐ Broccoli ☒ Shallots
☒ Carrot ☒ Bok Choy
☒ Onion ☐ Snow Peas
☐ Green Beans ☐ Sprouts

Choose up to 4 vegetables.
Each additional vegetable = \$0.50AUD

Choose Flavour: VERY Hot and spicy

Your Name: Customer Chris

☐ I'm feeling hungry ...
Make it a big serve!

CALCULATE

Added 5th vegetable. Total cost controls disappear.

Box of Noodles

Order Your Box of Noodles

Make your choices:

Noodle type:

☒ Rice Vermicelli ☐ Thin Egg ☐ Udon
☐ Wide Flat Rice ☐ Thick Egg ☐ Hokien

Add meat:

☐ Chicken \$5
☐ Pork \$5
☐ Beef \$5
☐ Seafood \$7
☒ None

Add vegetables:

☒ Capsicum ☐ Cabbage
☐ Broccoli ☒ Shallots
☒ Carrot ☒ Bok Choy
☒ Onion ☐ Snow Peas
☐ Green Beans ☐ Sprouts

Choose up to 4 vegetables.
Each additional vegetable = \$0.50AUD

Choose Flavour: VERY Hot and spicy

Your Name: Customer Chris

☐ I'm feeling hungry ...
Make it a big serve!

CALCULATE

\$6.00

☒ AUD
☐ USD

Calculate **Button** hit. Cost appears in AUD. (\$5.50 for noodles; \$0 for meat; \$0.50 for vegetables)

Box of Noodles

Order Your Box of Noodles

Make your choices:

Noodle type:

☒ Rice Vermicelli ☐ Thin Egg ☐ Udon
☐ Wide Flat Rice ☐ Thick Egg ☐ Hokien

Add meat:

☐ Chicken \$5
☐ Pork \$5
☐ Beef \$5
☐ Seafood \$7
☒ None

Add vegetables:

☒ Capsicum ☐ Cabbage
☐ Broccoli ☒ Shallots
☒ Carrot ☒ Bok Choy
☒ Onion ☐ Snow Peas
☐ Green Beans ☐ Sprouts

Choose up to 4 vegetables.
Each additional vegetable = \$0.50AUD

Choose Flavour: VERY Hot and spicy

Your Name: Customer Chris

☒ I'm feeling hungry ...
Make it a big serve!

CALCULATE

\$9.00

☐ AUD
☐ USD

Up-sized **CheckBox** checked. Total cost controls remain visible but cost increases by 50%.

Box of Noodles

Order Your Box of Noodles

Make your choices:

Noodle type:

☒ Rice Vermicelli ☐ Thin Egg ☐ Udon
☐ Wide Flat Rice ☐ Thick Egg ☐ Hokien

Add meat:

☐ Chicken \$5
☐ Pork \$5
☐ Beef \$5
☐ Seafood \$7
☒ None

Add vegetables:

☒ Capsicum ☐ Cabbage
☐ Broccoli ☒ Shallots
☒ Carrot ☒ Bok Choy
☒ Onion ☐ Snow Peas
☐ Green Beans ☐ Sprouts

Choose up to 4 vegetables.
Each additional vegetable = \$0.50AUD

Choose Flavour: VERY Hot and spicy

Your Name: Customer Chris

☒ I'm feeling hungry ...
Make it a big serve!

CALCULATE

\$6.93

☐ AUD
☒ USD

Currency changed to USD. Total cost is original AUD amount multiplied by 0.77.

Chicken added to order. Total cost controls disappear.

Noodle type changed and Calculate **Button** hit. Currency unchanged from previous selection. (Total cost calculation: \$5.50 for noodles; \$5 for meat; \$0.50 for vegetables + 50% for upsizing)