Second Assignment – Problem Solving Task

What's the Tip?

Weighting 20% - Due date: **19 June 2019**

Specification

You are to implement a simple tip calculator with a GUI interface - 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 acronyms mean, 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 location, size, colour etc of each control. Choose your own background. The look and feel of the widgets is your own choice. However, the functionality provided by your GUI must meet the specifications contained in this document.

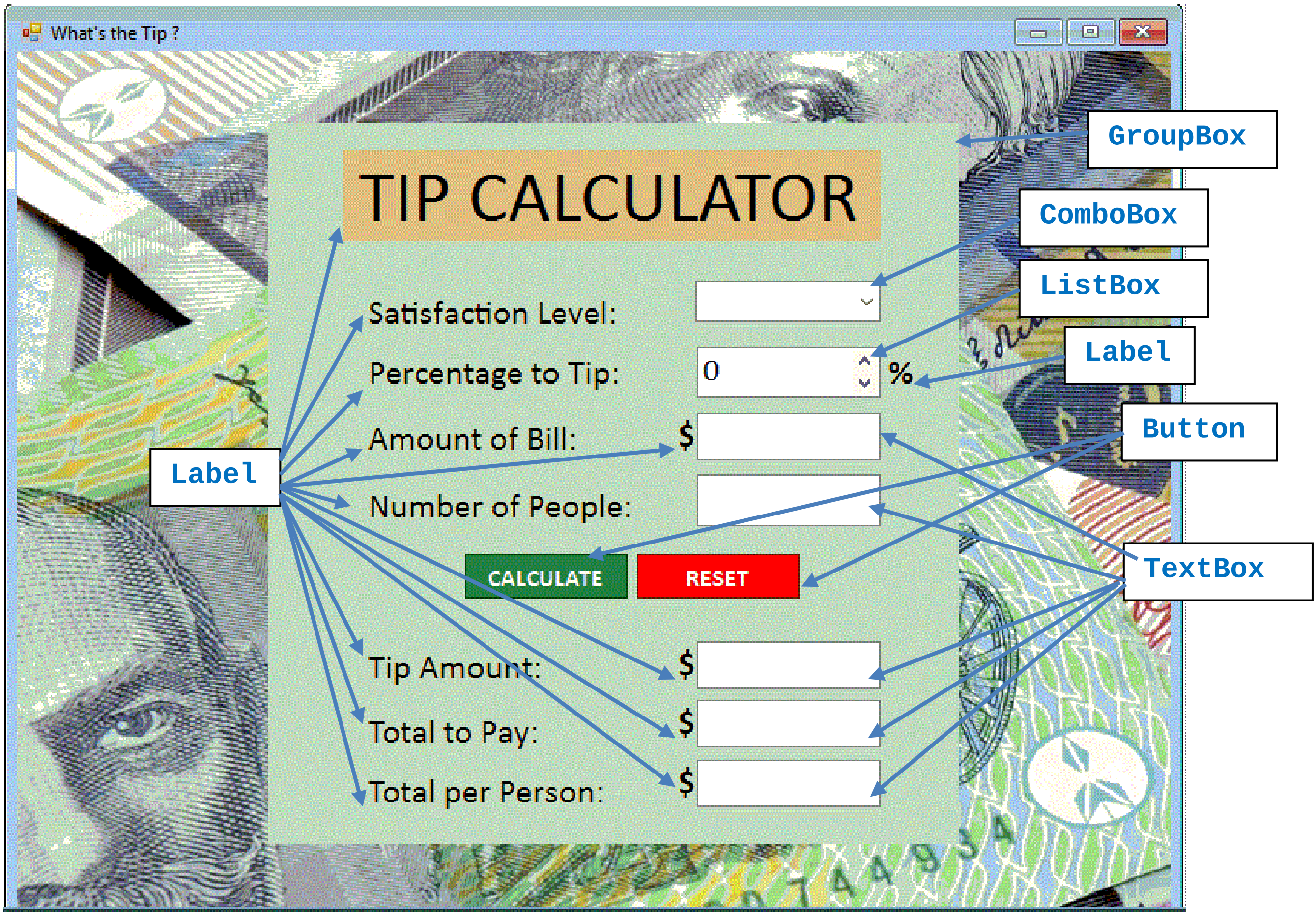


Figure 1: GUI Design

The GUI must consist of 13 `Label`s, 5 `TextBox`es, 2 `Button`s, 1 `GroupBox`, 1 `ComboBox`, and 1 `ListBox`. Not all controls will be visible or enabled at the start of the program.

The World of Tipping

Imagine going out for dinner with a group of friends and being presented at the end of the meal with a bill. It can often be a complex matter to sort out who is paying what. The easier way to resolve this is to split the bill equally between everyone, so that each pays the same amount. However, sometimes there are tips involved, which makes the task just a little harder.

A tip is an additional amount above the bill volunteered to be paid by the customer (eg for a meal) which reflects the user's overall appreciation or satisfaction level with the service. Greater satisfaction would warrant a larger tip. Displeasure with the service warrants no tip at all. In some cultures (eg USA) tips are expected for all services provided in the hospitality industry. Others are not expected at all (eg Australia), but of course are always very much welcomed! The amount of tips wildly varies between cultures.

Program Functionality

The program you are to write performs the cost calculations for a group of people, and displays an amount owing by each person. You will need to check that the user enters the correct TYPE of values, as well as values within an acceptable range (i.e. only positive values).

User Input

The program will allow for the user to enter several pieces of information.

- A satisfaction level. The options are chosen from a list which include: N/A, Perfect, Very Good, Good, Satisfactory and Poor. Each satisfaction level is to be associated with a tip percentage:

N/A	0%
Perfect	20%
Very Good	15%
Good	10%
Satisfactory	5%
Poor	0%

- The amount presented to the customer as the bill. This is the amount calculated by the service provider whatever services were supplied.
- The number of people in the group.

Output

When the necessary information has been entered correctly by the user, the following calculations are made and output displayed:

- Amount of the tip – a dollar value representing the tip (if any).
- Total amount to pay – a dollar value representing the bill plus the tip (if any).
- Total amount for each person to pay – a dollar value representing an equal proportion of the bill plus tip (if any), depending on how many people were involved.

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

When the program starts, note that:

- The Labels and ListBox associated with "**Percentage to Tip**" are not visible.
- The TextBoxes below the Buttons are (and remain) disabled.
- all other controls are as per the screenshot in Figure 2.
- Both Buttons should be visible and enabled.

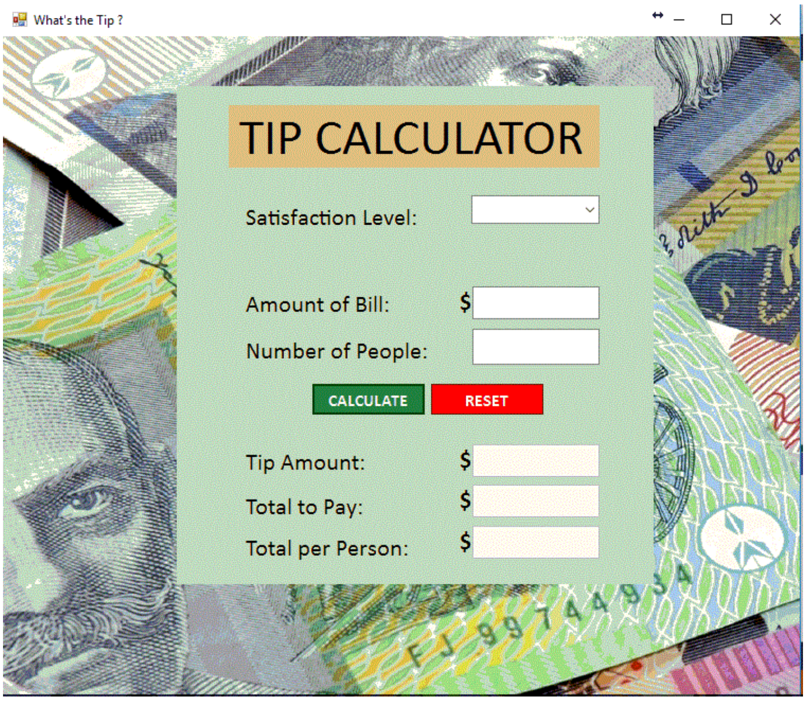


Figure 2: GUI at Startup

GUI after interaction

During interaction with the user, note that:

- When a satisfaction level is selected from the ComboBox, the associated tip percentage value appears in the Listbox below it. (See the previous section detailing the percentage associated with each satisfaction level). Hint: you can set the selected item in the ListBox using the index of the selected item in the ComboBox.
- Making a satisfaction level selection causes the TextBoxes under the Buttons to be cleared.
- The user can choose to ignore the satisfaction level (and hence the tip) by selecting **N/A**, in which case the Labels and ListBox associated with "**Percentage to Tip**" disappear from view (if they were visible prior to that choice).
- The ListBox associated with **Percentage to Tip** is (and remains) disabled so that the user cannot insert a value.
- When the user hits the **Calculate** Button, the values of the tip, total to pay and total per person are calculated and displayed in the appropriate places on the GUI.
- All amounts must be shown as floating point numbers to two decimal places.

- If the user hits the **Calculate** `Button` but has failed to enter a value in either of the `TextBox`es associated with the Amount of Bill or Number of People, a `MessageBox` is displayed to report this. See the Appendix at the end for examples.
- If a user enters a value of the incorrect *type*, a `MessageBox` should be displayed to report this and request the user change the value entered.
- If a user enters an unreasonable amount a `MessageBox` should be displayed to report this. An unreasonable amount would be, for example, a ***non-positive*** value for number of people, or a ***negative*** number for bill amount. $0 is acceptable for a bill amount (who wouldn't want that!?), but 0 people doesn't make sense.
- After an unreasonable amount has been dealt with (as described above), that value must be cleared from the appropriate `TextBox` so the user can enter another amount.
- When the user hits the **Reset** `Button` the `TextBox`es and `ComboBox` are all cleared and the widgets associated with the Percentage to Tip are hidden. In fact, the GUI returns to the state at Startup. See Figure 1.

General Requirements

Though in lectures 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 place any program logic into a separate method which is called from event handlers or other methods in your program.

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 discussed in class.

Your solution should be modularised ie broken into small, single-purpose methods (see Lecture 2). Event handler methods must *not* contain program logic (e.g., to make calculations), but instead should call other utility methods for this purpose.

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 ask or commission someone else to write the assignment for you, or help you write the assignment.

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

Final Comments

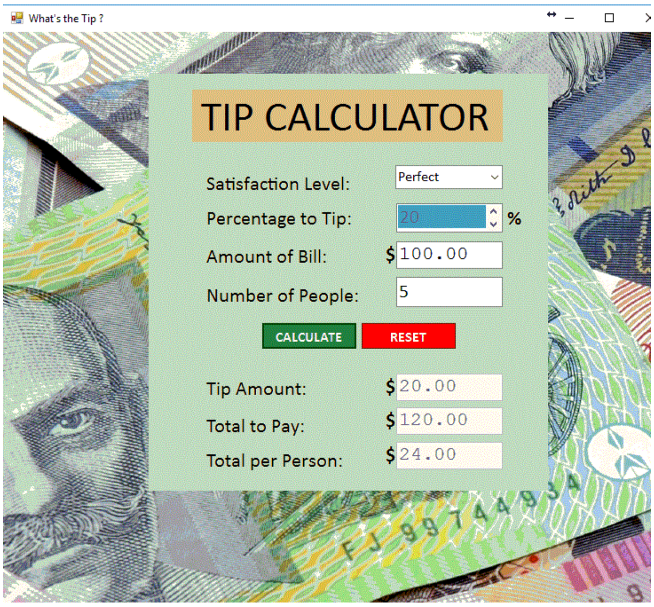
You should ensure that your project always compiles and does *something*.

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.

Plan to be finished well before the afternoon of the due date! You should avoid writing all of the code in one sitting and then attempt a build (compilation). Remember ***Incremental Development and Incremental Implementation.***

Happy tipping! Author: Donna Teague (2019)

Appendix - Sample screen shots after **Calculate** `Button` pressed:



TIP CALCULATOR

Satisfaction Level: Perfect

Percentage to Tip: 20 %

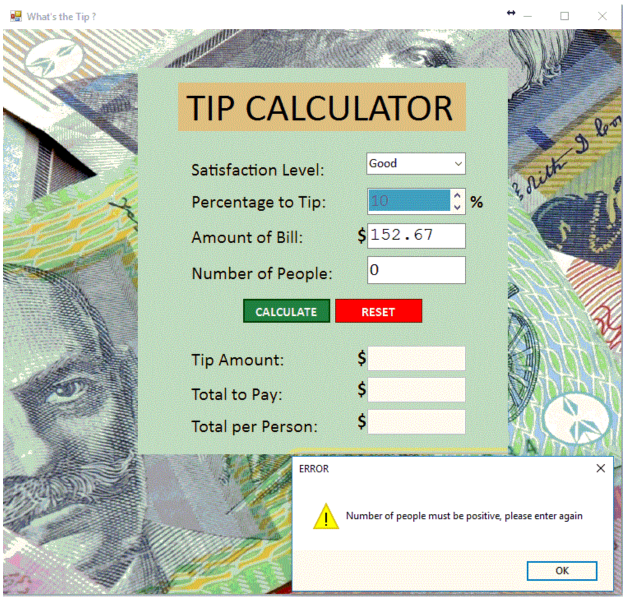
Amount of Bill: $ 100.00

Number of People: 5

Tip Amount: $ 20.00

Total to Pay: $ 120.00

Total per Person: $ 24.00



TIP CALCULATOR

Satisfaction Level: Good

Percentage to Tip: 10 %

Amount of Bill: $ 152.67

Number of People: 0

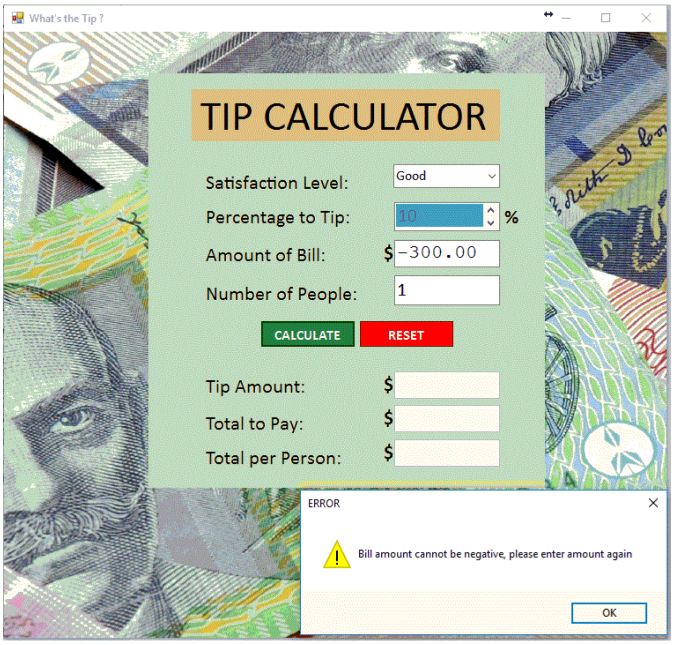
Tip Amount: $

Total to Pay: $

Total per Person: $

ERROR

Number of people must be positive, please enter again



TIP CALCULATOR

Satisfaction Level: Good

Percentage to Tip: 10 %

Amount of Bill: $ -300.00

Number of People: 1

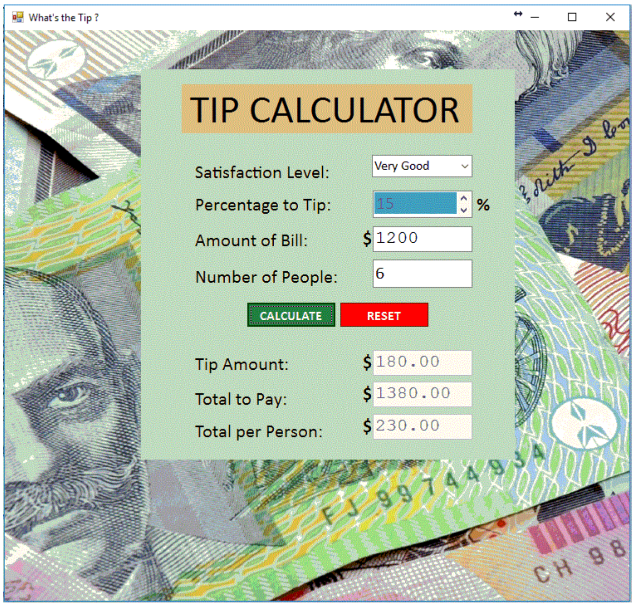
Tip Amount: $

Total to Pay: $

Total per Person: $

ERROR

Bill amount cannot be negative, please enter amount again



TIP CALCULATOR

Satisfaction Level: Very Good

Percentage to Tip: 15 %

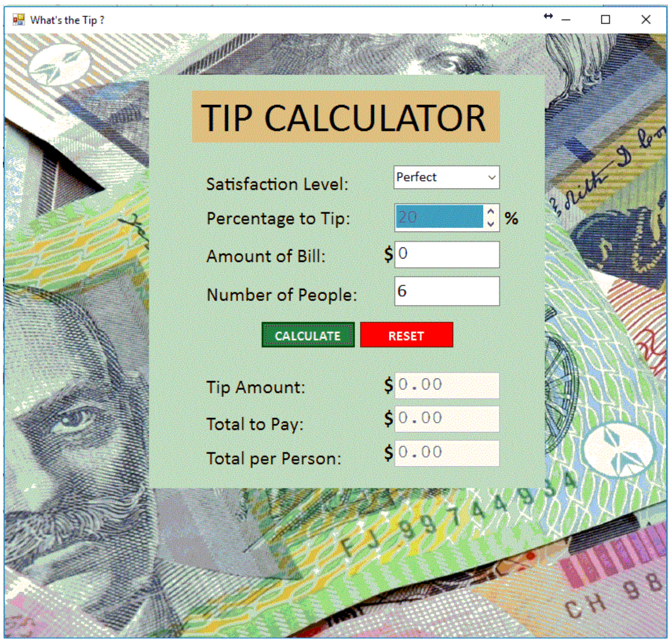
Amount of Bill: $ 1200

Number of People: 6

Tip Amount: $ 180.00

Total to Pay: $ 1380.00

Total per Person: $ 230.00



TIP CALCULATOR

Satisfaction Level: Perfect

Percentage to Tip: 20 %

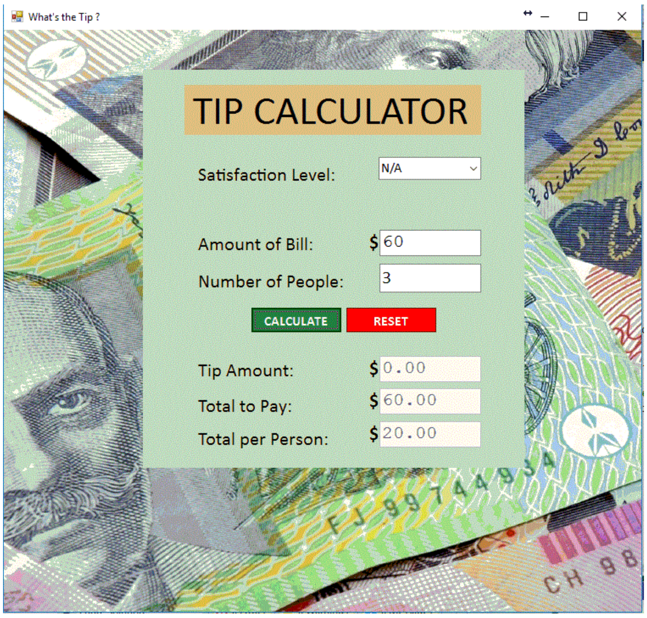
Amount of Bill: $ 0

Number of People: 6

Tip Amount: $ 0.00

Total to Pay: $ 0.00

Total per Person: $ 0.00



TIP CALCULATOR

Satisfaction Level: N/A

Amount of Bill: $ 60

Number of People: 3

Tip Amount: $ 0.00

Total to Pay: $ 60.00

Total per Person: $ 20.00