

# COMPX101 Practice Test 3

2018A

First Name/Given Name	Username
Surname/Family Name	ID Number

## Getting Started

To begin, download the template project from Moodle and unzip it. Open up the Visual Studio C# project which is inside this folder, and use this to create your solution.

## Instructions

This is an open book test: that is you may use your paper notes, Gaddis textbook, Moodle and Microsoft help. You may not use the Internet to search for answers or communicate with anyone. No USB devices/ drives, laptops, music devices or cellphones are allowed.

The test is **100** minutes long. The first **10** minutes are exclusively for writing the pseudocode for task 1 in the space provided. You then have a further **90** minutes to complete as much of the application as you can.

## Assessment

5	Created correct menu structure, with appropriate shortcut keys.
10	Pseudo-code completed for Open File menu option.
	<b>Part A</b>
5	Can use Open Dialog control to select filename (filters & cancel works).
5	File is opened and data is read from file & split correctly.
5	Data from file is written to Listbox correctly and neatly formatted.
5	Called the CalculateX method correctly.
10	Created and called the CalculateY method correctly.
10	Tower is drawn in the correct position.
10	Handles invalid data and displays the invalid data in the console window
5	Used constants appropriately.
	<b>Part B</b>
10	Created appropriate lists to store info and data is stored in lists correctly.
5	Created CountTowers method correctly.
5	Count Towers menu item calls method correctly and displays result.
10	Appropriate documentation as described in Task 12.

**Complete the pseudo-code for Open File menu item here**

Declare variables

Set the filter for the open file dialog control

IF user has selected a file to open THEN

Declare variables

Set the filter for the dialog control

IF user has chosen a file to open THEN

Open the chosen file for input

WHILE not end of file

Get type of object

Get the first string

Get the second string

Get the third number

Get the fourth number

Get the fifth number

Display the Contents of the File in the Listbox

Calculate the X and Y position

Draw Tower using Draw Tower Method

ENDIF

ENDWHILE

Close the file

ENDIF

ENDIF

## Problem Description

Vodafone wants to look at the coverage of their cellphone towers in Hamilton, and would like an application that lets them visualise this information. They have a CSV file of the cellphone towers in Hamilton, one tower per line in the following format: **Licensee, Location, Easting, Northing, Power**.

Here are the first three lines from the file **towers.csv**:

```
VODAFONE NZ LTD,HOOD STREET,2711200,6376600,27.4
VODAFONE NZ LTD,WAIKATO HOSPITAL,2711300,6375100,26.5
VODAFONE NZ LTD,RAILWAY STATION HN,2709700,6376500,21.8
```

The map is the NZMS260 map for Hamilton, and the Northing and Easting values give the position of the tower (how far North and how far East on the map of NZ). The Power is in kW.

You are to create an application which loads in a file of cellphone tower information and displays graphically the cellphone positions and coverage. It will have a picturebox showing a map to display the towers on (represented by small rectangles where each tower is located and circles over the map representing the coverage of the cellphone tower proportional in size to the power rating of the tower). It will also have a listbox to list the tower information and a textbox to enter a power value so that the application can count how many towers have a power rating less than or equal to the value the user types in. See example screenshots in the practical test folder on your desktop.

You must create a **File** menu with the following items:

```
Open File...
Count Towers
-
Exit
```

The **Open File** menu item will ask for the name of the input file and check that the user clicks on the OK button of the dialog control. It will then open the file and repeat the following until the end of file is reached. For each line, it will read in the line from the file, then split it and then extract the values from the split array into separate variables. It will then display the information on one line in the listbox, neatly padded so the information is in columns. It will then calculate the correct x and y position and then draw the tower (using the methods that have been provided).

## Tasks — Part A (80%)

1. Complete the pseudo-code for the Open File menu item on page 3.
2. Create the user interface to match the specifications above. The picturebox and listbox have been provided for you. Create the **File** menu as specified above assigning shortcut keys.
3. Write the menu event procedure for the **Exit** menu item which will exit from the application.
4. Write the click event method for the **Open File...** menu item which will set an appropriate filter for the **OpenFileDialog** control and then ask for the name of the CSV file containing the cellphone tower information.

It will then open the CSV file and read in one CSV line at a time until the end of the file is reached. You will need to extract the values from the CSV line and store the values into appropriate variables. For each line that you read, display the Location, Easting, Northing and Power values in the listbox, using the **PadRight** or **PadLeft** methods so that the values are in neat columns. After the end of the file has been reached the CSV file should be closed.

5. Modify your **Open File...** event method so that it checks for bad data in the file, such as the correct number of elements (commas) on a line, or the numbers parsing correctly. If it catches an exception then write the raw line to the console window but still carry on processing the next line. Test this using the **badtowers.csv** data file.
6. Modify your **Open File...** menu event method (step 4 above) so that as it reads each line of the file, it draws the cellphone tower on the map, as well as putting it into the ListBox. Each tower should be drawn as a small black rectangle at position (x,y), plus a circle with centre (x,y) and a radius equal to the power rating of the tower. Use the provided method to draw the tower.

How do you calculate the (x,y) position of each tower from the easting and northing values? Pass the Easting value of the tower to the CalculateX(...) method to calculate the x position, and just use 100 pixels for the y position for the moment (all the towers should appear along a horizontal line at y=100).

The provided **CalculateX(int easting)** method calculates this formula:

$$percentAcross = \frac{(easting - MinimumEasting)}{(MaximumEasting - MinimumEasting)}$$

$$x = pictureBoxWidth * percentAcross$$

7. Create a method called **CalculateY** that is passed the northing value and returns the correct Y position. Note that the Northing value indicates how far *up* the map the point is (from the bottom of the pictureBox), so the calculation is slightly different to the **CalculateX** method. The formula you need to implement is:

$$\text{percentUp} = \frac{(\text{northing} - \text{MinimumNorthing})}{(\text{MaximumNorthing} - \text{MinimumNorthing})}$$

$$y = \text{pictureBoxHeight} - (\text{pictureBoxHeight} * \text{percentUp})$$

*Hint: Make sure that the division is not performing integer division otherwise the y value will always be zero.*

Modify your **Open File...** menu event method so that it uses your **CalculateY** method to calculate the correct y position of each tower from its **Northing** value. A method to draw a tower has been provided for you, call it with the appropriate values. The tower positions should look similar to the example screenshot.

## Tasks — Part B (20%)

8. Write suitable (class scope) lists to store the important data about all the cellphone towers – the Location, Easting, Northing and Power values.
9. Modify the **Open File...** event method so that after extracting the data, the data is added to each list.
10. Create a method called **CountTowers** which is passed a power value as a double and returns back the number of towers which have a power rating less than or equal to the power value passed to the method.
11. Write the event method for the **Count Towers** menu item which gets a power value from the user and then displays the number of towers which have a power rating less than or equal to the given power value in a message window.
12. Document your application as stated below:
- You should have your name and ID as a comment at the top of your code.
  - Each click event method and your own methods should have a comment header describing what the method does. Type in 3 slashes '///' and C# will automatically put in a template comment header for the click event method. Then type in a description of the click event method in the summary section. Also document the parameters in your methods.
  - Appropriate inline comments in the code describing what the code is doing.