TOI-OHOMAI Institute of Technology **Semester Two, 2019** **Faculty of Engineering, Creative, Technology and Infrastructure.**	**Assignment Two** **Database and Visual Studio Forms** **COMP 4204 System Development** Given out: ***Week of 26th August 2019 (Week 6)*** Due Date: ***Part A Monday 9th September 2019*** ***Part B Monday 28th October 2019*** Marks: ***132*** Course Weighting: ***50%***

Aim

This assignment is intended to assess your skills in the following:

- Understand fundamental programming concepts and utilising relevant tools including: logic diagrams; text editor - command line or integrated development environment (IDE); compilers; interpreters.
- Develop applications including data models, databases and programming, with attention to user experience and documentation.
- Understand maths and logic concepts that underpin theory and practice of software development.
- Create a simple database including: creating tables, queries, forms and reports.

Instructions

There are two parts to this assignment:

Part 1. Create a new MS Access Database containing: Table, Form, Query and Report objects.

Part 2. Create a Windows Form application in MS Visual Studio 2015 to create a Celsius to Fahrenheit temperature converter.

Create a Windows Form application in MS Visual Studio 2015/2017 that will display the records from the tblContacts table of the AddressBook.mdb Database.

This assignment is an individual assignment.

You are required to upload a single zip file which contains all of the deliverables listed below for Part A of this assignment to Moodle no later than 4.00pm on Monday 9th September 2019. The assignment will not be accepted using any other method.

Part 2 is to be uploaded by Monday 25th October 2019.

Please clarify any aspect of the assignment of which you are unsure of with your lecturer.

Deliverables:

You will need to upload a zip file containing:

- **Part 1 – 2 x MS Access Files (Customer + Staff)** Mon 9th Sept
- **Part 2 – MS Visual Studio project folder** Mon 28th Oct

  *NOTE: The **project folder** must contain the solution file (.sln) and all of the files necessary to open the project.*

This zip file is to be uploaded to Moodle by selecting the Assignment 2 upload link.

PART 1 - Databases

Complete the following questions which will step you through the development of a new Database and the creation of various Database objects.

Questions One:

- Create a new Microsoft Access Database and name it **Customer**

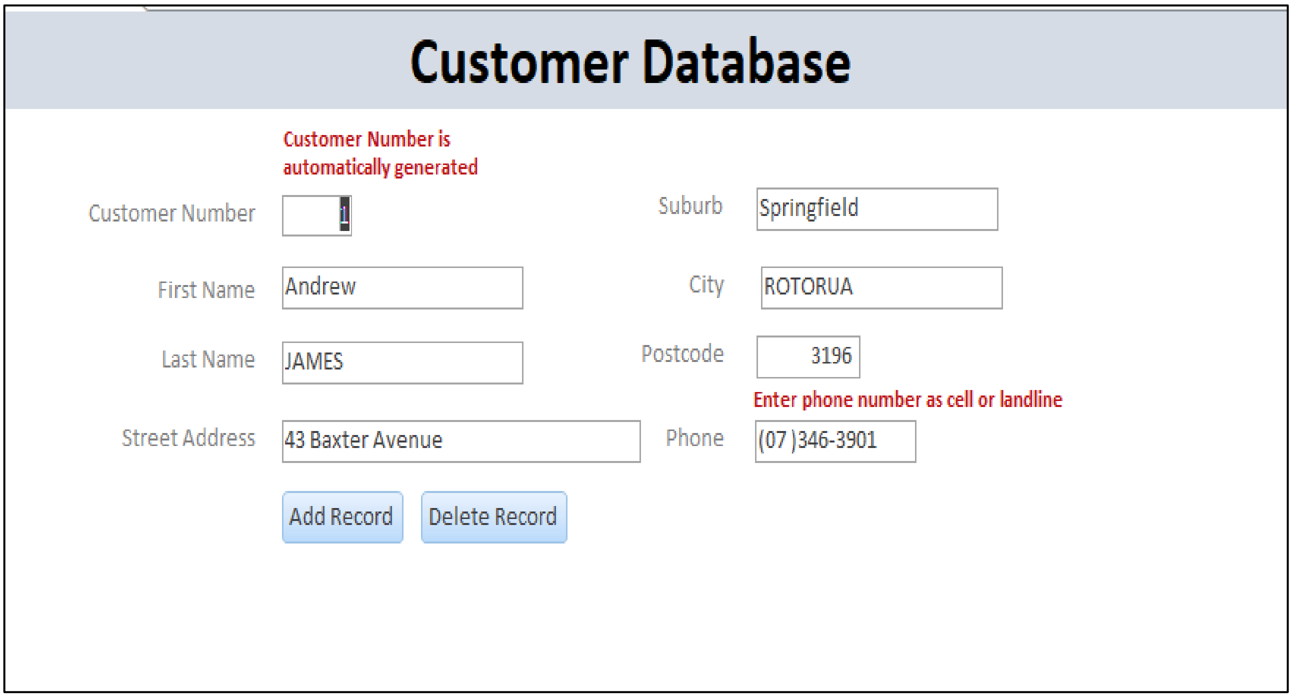
  *NOTE: The remainder of question one should be completed using the **Customer** Database created above.*

- Create a new table called **TblCustomers**

- In **Design View** of the TblCustomers table set up the following field names with the data types and field sizes given below: *Make sure you enter input masks and formatting as required.*

Field name	Data Type	Field Size	Input Mask/Format
CustNo	AutoNumber	Long Integer	
FName	Short Text	20	1st letter capital
LName	Short Text	20	Uppercase
StAddress	Short Text	30	
Suburb	Short Text	20	1st letter capital
City	Short Text	20	Uppercase
Postcode	Number	Integer, Fixed, 0dp	
Phone	Short Text	11	(009) 000-00099

- Create a new Form called **FrmCustomers** to look like the example below:



Customer Database

Customer Number is automatically generated

Customer Number: 1

First Name: Andrew

Last Name: JAMES

Street Address: 43 Baxter Avenue

Suburb: Springfield

City: ROTORUA

Postcode: 3196

Enter phone number as cell or landline

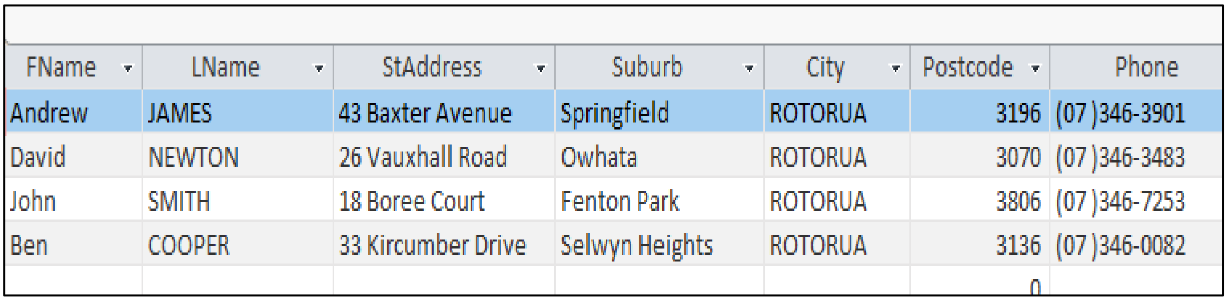
Phone: (07)346-3901

Ensure you apply the following formatting to the form:

- Add two instruction labels as shown and change the font size to 10pt and the colour to a dark red.
- Align the right-hand sides of all label boxes and left-hand sides of all text boxes
- Right align text on label boxes and left align text on text boxes – *see example above*
- Right align text within the customer number and postcode textboxes
- Expand all label text to full and add spaces where appropriate
(i.e. CustNo should be expanded to Customer Number)
- Format heading as shown

a. Add your name to the footer of the form

b. Using the Form called **FrmCustomers** populate the Database with the following records *correctly*:



FName	LName	StAddress	Suburb	City	Postcode	Phone
Andrew	JAMES	43 Baxter Avenue	Springfield	ROTORUA	3196	(07)346-3901
David	NEWTON	26 Vauxhall Road	Owhata	ROTORUA	3070	(07)346-3483
John	SMITH	18 Boree Court	Fenton Park	ROTORUA	3806	(07)346-7253
Ben	COOPER	33 Kircumber Drive	Selwyn Heights	ROTORUA	3136	(07)346-0082
					0	

Question Two:

a. Save the **Staff** Database from Moodle and rename it to **Employees.**

NOTE: The remainder of question two should be completed using the Staff Database you saved above.

b. Create a new **Query** (based on the table tblStaff) to show all staff members with a tax code of **S.**

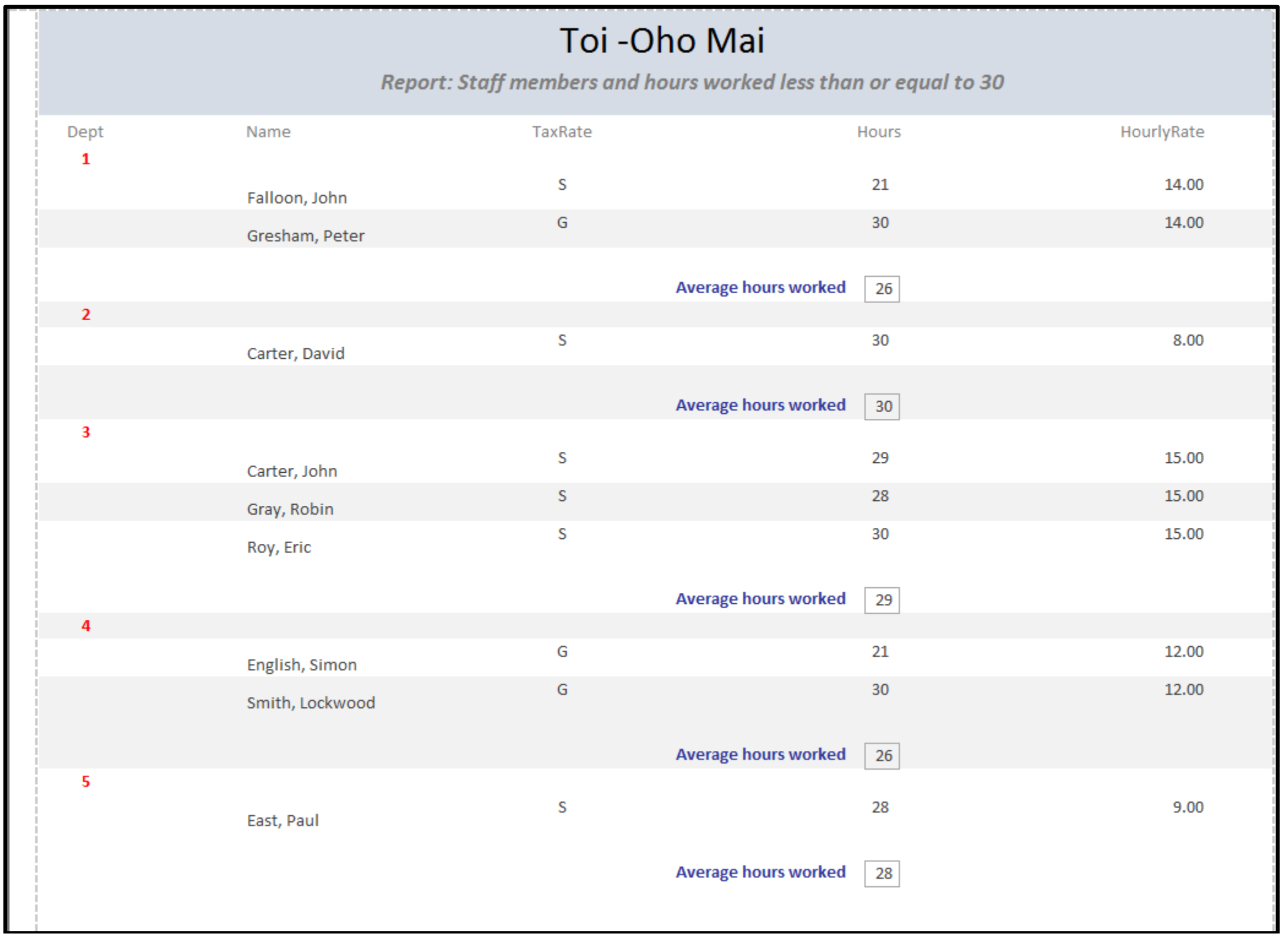
- Include ALL fields
- Name the query **QrySTaxCode**.
- Build a new calculated field and call it **Gross** (Hours multiplied by Rate)
- Build a new calculated field and call it **Tax** (30% of Gross)
- Format the both the Gross and Tax fields as **standard, 2dp**

c. Create a new **Query** (based on the table tblStaff) to show all staff members that have worked hours **less than or equal to 30.**

- Include **ALL fields**
- Name the query **QryHours**

d. Create a new **Report** displaying the staff members that have worked hours less than or equal to 30.

- Include ALL fields *except* **StaffID**
- *Group* by **Dept**
- *Sort* by **Name** in alphabetical order
- Display the *Average* of the **Hours** field
- Name the report **RptHours**
- Put the name of the company **Toi -Ohomai** in the heading and a description of what the report is showing
- Add your name to the footer
- Remove Summary label and tidy up layout
- Align Data so spaces between columns are even and all data is visible.
- Centre align **Dept** and **TaxCode** labels with their data
- Change the label **Avg** to read **Average hours worked** and move the label to sit next to the value.



Toi -Oho Mai

Report: Staff members and hours worked less than or equal to 30

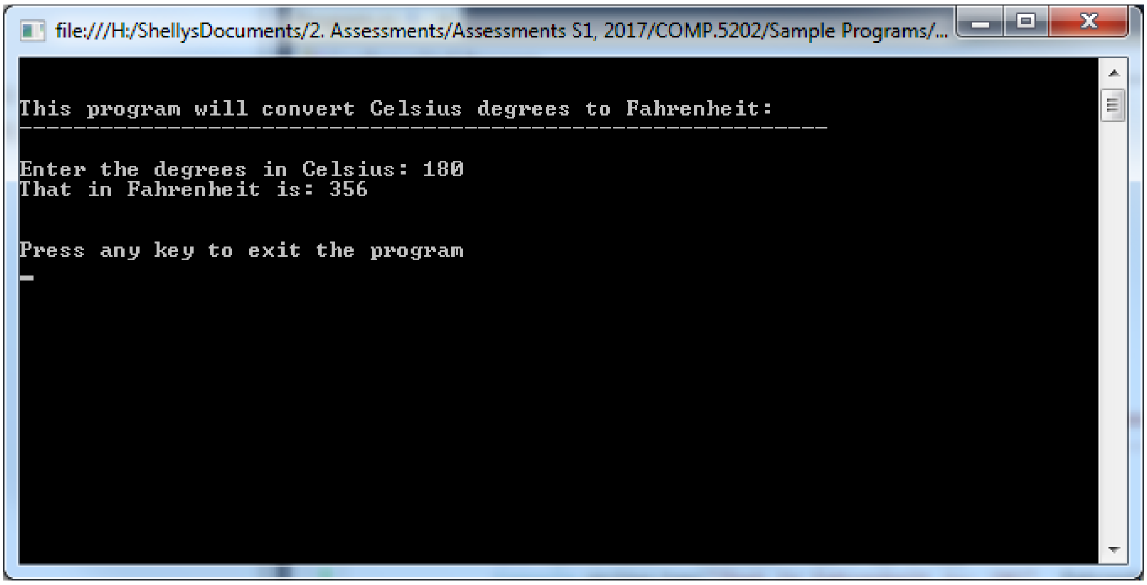
Dept	Name	TaxRate	Hours	HourlyRate
1				
	Falloon, John	S	21	14.00
	Gresham, Peter	G	30	14.00
		Average hours worked	26	
2				
	Carter, David	S	30	8.00
		Average hours worked	30	
3				
	Carter, John	S	29	15.00
	Gray, Robin	S	28	15.00
	Roy, Eric	S	30	15.00
		Average hours worked	29	
4				
	English, Simon	G	21	12.00
	Smith, Lockwood	G	30	12.00
		Average hours worked	26	
5				
	East, Paul	S	28	9.00
		Average hours worked	28	

PART 2 - Programming

Question One

Below is an example of a console application which converts Celsius degrees to Fahrenheit.

CONSOLE EXAMPLE:



file:///H:/ShellysDocuments/2. Assessments/Assessments S1, 2017/COMP.5202/Sample Programs/...

```
This program will convert Celsius degrees to Fahrenheit:
------------------------------------------------------------

Enter the degrees in Celsius: 180
That in Fahrenheit is: 356


Press any key to exit the program
_
```

For question one you are to develop this program but **NOT as a Console application**, as a **Windows Forms application** instead, in MS Visual Studio 2017.

You are to use textboxes for your input and output, with clear labels alongside each.

Validate your input by checking if the Celsius has been entered and that it is a number.

A Button_Click event will trigger the calculation and output the result in a textbox. This textbox should be read only, with no tab stop.

The formula for the conversion is: ***Fahrenheit = Celsius x 18 / 10 + 32***

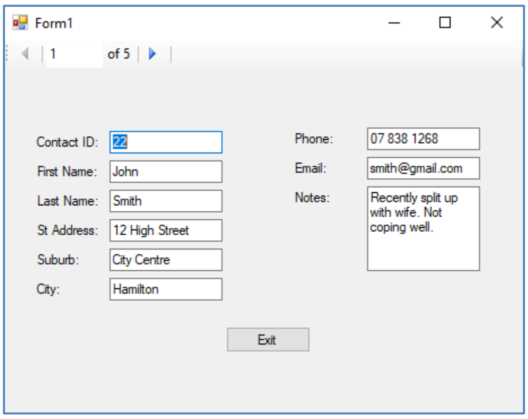
Display the name of the program in the title bar of the form.

Question Two

- Create a Windows Form application and call it **WindowsAddBookApp**
- On the form, display the data from the **tblContacts** table of the **AddressBook.mdb** Database

NOTE: *You may use the Data source configuration wizard to create the connection to the Database.*

- Ensure the records are displayed in **Detail**, not DataGridView
- Change the **Last Name** control type from a textbox to a ComboBox, so the user is able to select the Last name from a drop-down list.
- Edit the Form text property so an appropriate name for the program is displayed on the title bar.
- Arrange the labels and textboxes, so that **Phone**, **Email** and **Notes** fields are on the right-hand side of the form and the other fields are on the left.
- Make sure the text within all labels are written in full.
- Edit the labels and textboxes properties to all have a font size of 10pt
- Make the **Notes** textbox multiline and ensure it is big enough to display the data.
- Make all other textboxes an appropriate size for their data.
- Ensure the form is of an appropriate size, with equal space to the left, right, top and bottom of the objects.



Form1

1 of 5

Contact ID: 22

First Name: John

Last Name: Smith

St Address: 12 High Street

Suburb: City Centre

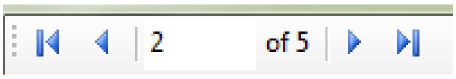
City: Hamilton

Phone: 07 838 1268

Email: smith@gmail.com

Notes: Recently split up with wife. Not coping well.

- Ensure all objects on the form are spaced proportionally
- Edit the Binding Navigator to show the Navigation options only, as shown below:



- Add an **Exit** button to the bottom of the form. When the user clicks on this button a dialogue box should appear asking the user “Are you sure you want to exit the program?” If the user clicks on the Yes option, the program will close without errors, if the user clicks on the No option, the program will return to the main Form.
- Add a splash screen to your application. This splash screen should display an appropriate image and information about you, the developer.

Part One – Databases **Name** **Date**

MARKING SCHEME

Question One – Create a Database and call it **CUSTOMER**				**1**	
TABLE					
Create a Table and call it **tblCustomers**				**2**	
In design view set the field names and input masks					
Field Name	**Data Type**	**Field Size**	**Input mask**		
CustNo	AutoNumber	Long Integer		**2**	
FName	Short Text	20	1st letter capital	**2**	
LName	Short Text	20	Uppercase	**2**	
StAddress	Short Text	30		**2**	
Suburb	Short Text	20	1st letter capital	**2**	
City	Short Text	20	Uppercase	**2**	
Postcode	Number	Integer, Fixed, 0dp		**2**	
Phone	Short Text	11	(009) 000-00099	**2**	
FORM – See example on page 2				**2**	
Add two instruction labels as per example and change font size to 10pts and make dark red				**2**	
Alignment of label boxes				**2**	
Right align text within all labels				**2**	
Right align text within the customer number and Postcode textboxes				**2**	
Expand all label text in full i.e. CustNo should be Customer Number				**2**	
Add name to footer of form				**2**	
Add a delete and an add button to your form				**2**	
Populate form				**5**	
				40 marks	

Question Two – Open **Staff** database from Moodle and save it as **Employees**	**1**	
Queries		
Query One – Save as **QryStaxCode**. Include all fields – All staff with a Tax code of S	**6**	
Calculated field called Gross – Hours X Rate	**3**	
Calculated field called Tax – 30% of Gross	**3**	
Format them both as Standard 2dp	**2**	
Query Two – Save as **QryHours**. Include all fields. All staff that have worked hours less than or equal to 30	**6**	
	21 Marks	

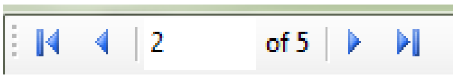
Report		
Base the report on Query 2 – Staff that have worked less than 30 hours – Name the Report **RptHours**	**2**	
Include all fields except **StaffID**	**1**	
Group by **Dept**	**2**	
Sort by **Name** in Alphabetical order	**2**	
Display the **Average** of the **Hours** field	**3**	
Put a **label** in the **heading** named **Toi Ohomai** and a **description** of what the report is showing	**2**	
Add your **name** to the **Footer**	**2**	
Remove **Summary Label** and tidy up layout	**2**	
Align data – all spaces even and all data visible	**1**	
Centre align **Dept** and **TaxCode** labels with their data.	**1**	
Change the label Avg to read **Average hours worked** and move it to sit beside the value	**1**	
	19 marks	

PART 2 - Programming

Question 1

Entire project folder is submitted (including .sln)	**1**	
Windows Form application created, layout is appealing, objects are placed proportionally on the form	**3**	
At least 1 variable declared to store Celsius	**1**	
Label prompts user to enter Celsius	**1**	
User able to enter Celsius via textbox	**1**	
If statement used; to check if textbox is empty and input is numeric.	**3**	
Button click event coded:	**1**	
Fahrenheit calculated, correctly	**2**	
Result is displayed, via textbox, Read only = true, tab stop = false	**4**	
Label informs user of what is displayed in result textbox	**1**	
Form text property changed, to contain program name	**2**	
Comments have been added to code (includes program brief and name of developer)	**1**	
Program runs without errors	**1**	
Question One (Celsius to Fahrenheit) Total	**22**	

Question 2

Create a Windows Form application and call it **WindowsAddBookApp**	**2**	
On the form, display the data from the **tblContacts** table of the **AddressBook.mdb** Database ***NOTE:** You may use the Data source configuration wizard to create the connection to the Database.*	**2**	
Ensure the records are displayed in **Detail**, not DataGridView.	**1**	
Change the **Last Name** control type from a textbox to a ComboBox, so the user is able to select the Last name from a drop-down list.	**2**	
Edit the Form text property so an appropriate name for the program is displayed on the title bar.	**1**	
Arrange the labels and textboxes, so that **Phone**, **Email** and **Notes** fields are on the right-hand side of the form and the other fields are on the left.	**2**	
Make sure the text within all labels are written in full.	**1**	
Edit the labels and textboxes properties to all have a font size of 10pt	**2**	
Make the **Notes** textbox multiline and ensure it is big enough to display the data.	**2**	
Make all other textboxes an appropriate size for their data.	**1**	
Ensure the form is of an appropriate size, with equal space to the left, right, top and bottom of the objects.	**2**	
Ensure all objects on the form are spaced proportionally	**1**	
Edit the Binding Navigator to show the Navigation options only, as shown below: 	**1**	
Add an **Exit** button to the bottom of the form. When the user clicks on this button a dialogue box should appear asking the user "Are you sure you want to exit the program?" If the user clicks on the Yes option, the program will close without errors, if the user clicks on the No option, the program will return to the main Form.	**4**	
Add a splash screen to your application. This splash screen should display an appropriate image and information about you, the developer.	**6**	
	30	

WARNING

It is your responsibility to ensure that all of the correct files, with the correct filenames are contained in one single zip file and are submitted to Moodle by the due date.

Ensure that your program for part 2 runs without errors in a Toi –Ohomai laboratory (ie: must be Microsoft Visual Studio 2015 version). Failure to meet any of these requirements may mean that you lose marks. The assignment must be a product of **your own** work except for use of resources supplied with the course, discussions conducted by the lecturer during class time, and other assistance shown as acceptable in the section *Assistance to Other Students.*

Marking

Your lecturer may/will use an oral examination to test your understanding of the material submitted.

Assistance to other Students

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another's learning and also what types of assistance are unacceptable in an assessment.

Beneficial Assistance

- Study Groups.
- Discussion.
- Sharing reading material.
- Testing another student's programming work using the executable code and giving them the results of that testing.

Unacceptable Assistance (will not be tolerated)

- Working together on one copy of the assessment and submitting it as own work.
- Giving another student your work.
- Copying someone else's work. This includes work done by someone not on the course.
- Changing or correcting another student's work.
- Copying from books, Internet etc. and submitting it as own work. Anything taken directly from another source must be acknowledged correctly: Show the source alongside the quotation.

REMEMBER – Your lecturer will most probably ask you questions on certain aspects of your practical work – the expectation would be "If you did the work then you can answer the questions". Failure not to be able to answer said oral questions would indicate that you are not the owner of the assignment.