Unitec Institute of Technology
TE WHARE WĀNANGA O WAIRAKA

Department of Computing

ISCG6421 – GUI Programming
Semester 1, 2015

Assignment 2

Deadline Time:	**8:30 a.m.**
Deadline Date:	**Monday 13th April 2015**
Course Weighting:	**40%**
Marks:	**100**

Assignment objectives:

There are two objectives for this assignment:

1. To demonstrate the use of the C# language for the creation of an effective database front end by developing forms which:
    a. Visually present database information.
    b. Demonstrate appropriate user and programmatic navigation.
    c. Allow the user to enter new data and change existing data.
2. To demonstrate the testing, debugging and documentation of a C# application by:
    a. Using the available debugging tools to debug a program.
    b. Developing test cases appropriate to a GUI approach.

Introduction

This assignment is based on a case study on the Big Eye Private Detective Agency. The agency requires a front end application for their database which has already been developed. You, the programmer, have to complete the work by writing an application, in the language C#, which will allow the users to maintain the records stored in the database and produce the report that they need.

Database

You are provided with an MS Access database for this assignment named **Big Eye.**
Please leave the structure of this database as it is and not add any tables or queries.

The Big Eye database has the following tables:

CLIENT (ClientID, LastName, FirstName, Street Address, Suburb, City, PhoneNumber)
INVESTIGATOR (InvestigatorID, LastName, FirstName, HourlyRate, CellPhone, Licensed)
CASE (CaseID, Description, Status, CaseType, DateOpened, ClientID*)
ASSIGNMENT (CaseID*, InvestigatorID*, Hours, DateAssigned)
EQUIPMENT (EquipmentID, Description, InvestigatorID*)

Note that the primary keys are underlined and foreign keys are indicated by a '*' symbol.

Guide to the Assignment

- You are to write the code for Assignment 2 in C# using **Microsoft Visual Studio 2010**
- You are to use the **OleDb** objects to interact with the **MS Access** database tables.
- Make sure that your C# project database connection points to **c:\temp** and make sure the **MS Access** database file is placed in **c:\temp**
- You are to use the shared dataset model (see the **SharedData** and **GlendeneCatCare** projects, as discussed in class, and notes on databases in C#).
- You are to use programming logic (master-detail relationships in conjunction with 'for-each' loops) to create the report – please do **not** use Crystal Reports.
- Your interface should conform to the standards discussed in class – please note the interface in **GlendeneCatCare** has limitations and should be used as a starting point but not as an exemplar.
- Use the metadata provided in **MS Access** database tables as the data dictionary from which you can make interface design decisions.
- Your test cases must include all of the tests that as listed as required in the assignment document and your tests must be repeatable and independent. You may use the data in the MS Access tables as the basis for your loaded test data.
    - Select your input test data to make your tests as independent as possible. For example: if you are testing that a program deletes a stage record then use, as test data, a stage that exists in the loaded test data NOT a stage that was input in an earlier test.
    - Remember to verify the expected results of the tests with other tests if possible. For example, if you have a test that adds a new record, then use the 'view' facility in the program to check that the record has been added.
    - The numbering of your tests MUST correspond with the numbers in this document. If you create your interface so that a test is redundant (e.g. you use radio buttons for selection) you must still include the test in your test cases along with an explanation as to why it is redundant.

Test Cases
Please using the following format for your test cases

Requirement to test	Test Data Input	Expected Outcomes	Actual Outcomes

Deadline Date
Please note that the first deadline date is a week prior to the demonstration date. The reason for this deadline is due to the policy regarding late submissions.

Application Requirements

Write a Windows application in C# that allows the user to do the tasks specified below and to provide comprehensive testing documentation that proves that the application performs all of the tasks. Please note that these requirements are non-negotiable

Task 1: Main Menu

- This task allows the user to access all the forms and the report in the system and exit the application.
- When run, this task will create a data module object (containing the OleDb objects used with the MS Access database tables) the address of which will be passed on to all the other forms via a parameter.

Required tests:

1.1. Main menu is displayed when the program is run.
1.2. Display client maintenance form when the client maintenance button is clicked
1.3. Display investigator maintenance form when the investigator maintenance button is clicked
1.4. Display equipment maintenance form when the equipment maintenance button is clicked
1.5. Display case maintenance form when the case maintenance button is clicked
1.6. Display case assignment maintenance form when the case assignment maintenance button is clicked
1.7. Display the report “Invoices” form when the report “Invoices” button is clicked
1.8. Exit the program when the exit button is clicked.

Task 2: Report “Invoices”

- This must allow the user to print (on a print form) a report called ‘Invoices’ that shows the client ID, full name, full address, case name, date opened and gross due of each open case. The gross due of a case is the sum of the assignments for that case (hours multiplied by the investigator’s hourly rate). All currency values must be formatted (e.g. 10 must be shown as $10.00 and right-justified). One client per page. Please note that Crystal Report is **NOT** to be used for this task and the report is to be produced using Master-Detail relationships in conjunction with ‘for-each’ loops.

Required tests:

2.1. Produce the report with all requested fields present, correct calculations and for open cases only

Task 3: Client Maintenance

- Display the Clients' names in a list.
- Allows the user to navigate between Clients in the list.
- Display a selected Client's details in a read-only format.
- If the user clicks on the **Add Client** button then
    - The **Modify Client** and **Delete Client** buttons are disabled.
    - A panel should appear (showing **Save Client** and **Cancel** buttons) allowing the user to enter new values for the Client's last name, first name, street address, suburb, city and phone number.
    - If the user enters valid data for all fields and clicks on the **Save Client** button then a new Client record is saved in the database.
    - If the user clicks on the **Cancel** button then panel should disappear and the **Modify Client** and **Delete Client** buttons enabled.
- If the user clicks on the **Modify Client** button then
    - The **Add Client** and **Delete Client** buttons are disabled.
    - A panel should appear (showing **Update Client** and **Cancel** buttons) allowing the user to change the values for the Client's last name, first name, street address, suburb, city or phone number.
    - If the user makes valid changes to any of the allowable fields and clicks on the **Update Client** button then the Client record is updated in the database.
    - If the user clicks on the **Cancel** button then the panel should disappear and the **Add Client** and **Delete Client** buttons enabled.
- If the user clicks on the **Delete Client** button and the selected Client has no cases then the user is asked to confirm the deletion – if the user confirms the deletion then the Client is deleted
- If the user clicks on the **Delete Client** button and the selected Client has cases then the error message "You may only delete Clients who have no cases" is displayed.
- If the user clicks on the **Return** button then focus is returned to the Main Menu.

Required tests:

3.1. Add a new Client with valid data in all fields
3.2. Modify an existing Client's last name with valid data
3.3. Modify an existing Client's first name with valid data
3.4. Modify an existing Client's street address with valid data
3.5. Modify an existing Client's suburb with valid data
3.6. Modify an existing Client's city with valid data
3.7. Modify an existing Client's phone number with valid data
3.8. Delete an existing Client who has no cases
3.9. Return to main menu
3.10. Attempt to add a new Client with an invalid last name but with valid data in all other fields
3.11. Attempt to add a new Client with an invalid first name but with valid data in all other fields
3.12. Attempt to add a new Client with an invalid street address but with valid data in all other fields
3.13. Attempt to add a new Client with an invalid suburb but with valid data in all other fields
3.14. Attempt to add a new Client with an invalid city but with valid data in all other fields
3.15. Attempt to add a new Client with an invalid phone number name but with valid data in all other fields
3.16. Attempt to modify an existing Client with an invalid last name
3.17. Attempt to modify an existing Client with an invalid phone number
3.18. Attempt to delete an existing Client who has cases

Task 4: Investigator Maintenance

- Display the Investigators' names in a list.
- Allows the user to navigate between Investigators in the list.
- Display a selected Investigator's details in a read-only format.
- If the user clicks on the **Add Investigator** button then
    - The **Modify Investigator** and **Delete Investigator** buttons are disabled.
    - A panel should appear (showing **Save Investigator** and **Cancel** buttons) allowing the user to enter new values for the Investigator's last name, first name, hourly rate, cell phone and licensed status.
    - If the user enters valid data for all fields and clicks on the **Save Investigator** button then a new Investigator record is saved in the database.
    - If the user clicks on the **Cancel** button then the panel should disappear and the **Modify Investigator** and **Delete Investigator** buttons enabled.
- If the user clicks on the **Modify Investigator** button then
    - The **Add Investigator** and **Delete Investigator** buttons are disabled.
    - A panel should appear (showing **Update Investigator** and **Cancel** buttons) allowing the user to change the values for the Investigator's last name, first name, hourly rate, cell phone or licensed status (using a checkbox).
    - If the user makes valid changes to any of the allowable fields and clicks on the **Update Investigator** button then the Investigator record is updated in the database.
    - If the user clicks on the **Cancel** button then panel should disappear and the **Add Investigator** and **Delete Investigator** buttons enabled.
- If the user clicks on the **Delete Investigator** button and the selected Investigator has no assignments or equipment then the user is asked to confirm the deletion – if the user confirms the deletion then the Investigator is deleted.
- If the user clicks on the **Delete Investigator** button and the selected Investigator has assignments then the error message "You may only delete Investigators who are not assigned to cases" is displayed.
- If the user clicks on the **Delete Investigator** button and the selected Investigator has equipment then the error message "You may only delete Investigators who are not allocated equipment" is displayed.
- If the user clicks on the **Return** button then focus is returned to the Main Menu.

Required tests:

4.1. Add a new Investigator with valid data in all fields
4.2. Modify an existing Investigator's last name with valid data
4.3. Modify an existing Investigator's first name with valid data
4.4. Modify an existing Investigator's hourly rate with valid data
4.5. Modify an existing Investigator's cell phone with valid data
4.6. Delete an existing Investigator who has no assignments and no equipment
4.7. Return to main menu
4.8. Attempt to add a new Investigator with an invalid last name but with valid data in all other fields
4.9. Attempt to add a new Investigator with an invalid first name but with valid data in all other fields
4.10. Attempt to add a new Investigator with an invalid hourly rate but with valid data in all other fields
4.11. Attempt to add a new Investigator with an invalid cell phone but with valid data in all other fields
4.12. Attempt to add a new Investigator with an invalid licensed status but with valid data in all other fields
4.13. Attempt to modify an existing Investigator with an invalid last name
4.14. Attempt to delete an existing Investigator who has assignments but no equipment
4.15. Attempt to delete an existing Investigator who has equipment but no assignments

Task 5: Equipment Maintenance

- Display the descriptions of the Equipment in a list.
- Allows the user to navigate between Equipment in the list.
- Display a selected item of Equipment's details in a read-only format.
- If the user clicks on the **Add Equipment** button then
    - The **Modify Equipment** and **Delete Equipment** buttons are disabled.
    - A panel should appear (showing **Save Equipment** and **Cancel** buttons) allowing the user to enter a new value for the Equipment's description (the investigator ID is set to null).
    - If the user enters valid data for the description and clicks on the **Save Equipment** button then a new Equipment record is saved in the database.
    - If the user clicks on the **Cancel** button then the panel should disappear and the **Modify Equipment** and **Delete Equipment** buttons enabled.
- If the user clicks on the **Modify Equipment** button then
    - The **Add Equipment** and **Delete Equipment** buttons are disabled.
    - A panel should appear (showing **Update Equipment** and **Cancel** buttons) allowing the user to change the values for the Equipment's description and investigator (using a combo box).
    - If the user makes valid changes to any of the allowable fields and clicks on the **Update Equipment** button then the Equipment record is updated in the database.
    - If the user clicks on the **Cancel** button then panel should disappear and the **Add Equipment** and **Delete Equipment** buttons enabled.
- If the user clicks on the **Delete Equipment** button and the selected Equipment is not assigned to an investigator then the user is asked to confirm the deletion – if the user confirms the deletion then the Equipment record is deleted
- If the user clicks on the **Delete Equipment** button and the selected Equipment is assigned to an investigator then the error message "You may only delete Equipment that is not assigned to an investigator" is displayed
- If the user clicks on the **Return** button then focus is returned to the Main Menu.

Required tests:

5.1. Add new Equipment with valid data in all fields
5.2. Modify existing Equipment description with valid data
5.3. Modify existing Equipment investigator with valid data
5.4. Delete existing Equipment that is not assigned to an investigator
5.5. Return to main menu
5.6. Attempt to add new Equipment with an invalid description but with valid data in all other fields
5.7. Attempt to delete existing Equipment that is assigned to an investigator
5.8. Attempt to modify existing equipment description with invalid data
5.9. Attempt to add new equipment with an invalid description but with valid data in all other fields
5.10. Attempt to delete existing equipment that is allocated to an investigator

Task 6: Case Maintenance

- Display the Cases' ids in a list.
- Allows the user to navigate between Cases in the list.
- Display a selected Case's details (along with the client's full name) in a read-only format.
- If the user clicks on the **Add Case** button then
    - The **Modify Case** and **Delete Case** buttons are disabled.
    - A panel should appear (showing **Save Case** and **Cancel** buttons) allowing the user to enter new values for the Case's description, client (using a combo box), type (using a combo box) and date opened (using a date picker). All new Cases are to have a status of Open.
    - If the user enters valid data for all fields and clicks on the **Save Case** button then a new Case record is saved in the database.
    - If the user clicks on the **Cancel** button then the panel should disappear and the **Modify Case** and **Delete Case** buttons enabled.
- If the user clicks on the **Modify Case** button and the selected Case has a status of Open then
    - The **Add Case** and **Delete Case** buttons are disabled.
    - A panel should appear (showing **Update Case** and **Cancel** buttons) allowing the user to change the values for the Case's description, type (using a combo box), or date opened (using a date picker).
    - If the user makes valid changes to any of the allowable fields and clicks on the **Update Case** button then the Case record is updated in the database.
    - If the user clicks on the **Cancel** button then panel should disappear and the **Add Case** and **Delete Case** buttons enabled.
- If the user clicks on the **Modify Case** button and the selected Case has a status of Closed then the message "Cannot update a closed Case" is displayed.
- If the user clicks on the **Delete Case** button and the selected Case has a status of Closed then the user is asked to confirm the deletion – if the user confirms the deletion then the Case record is deleted.
- If the user clicks on the **Delete Case** button and the selected Case has a status of Open then the error message "You may only delete closed Cases" is displayed.
- If the user clicks on the **Mark Case as Closed** button and the selected Case has a status of Open then all Case assignment records for the Case are deleted and the Case's status is changed to Closed.
- If the user clicks on the **Mark Case as Closed** button and the selected Case has a status of Closed then the error message "The Case is already closed" is displayed.
- If the user clicks on the **Return** button then focus is returned to the Main Menu.

Required tests:

6.1. Add a new case with valid data in all fields
6.2. Modify an existing open case's description with valid data
6.3. Modify an existing open case's case type with valid data
6.4. Modify an existing open case's status with valid data
6.5. Modify an existing open case's date opened with valid data
6.6. Delete an existing case that is closed
6.7. Return to main menu
6.8. Attempt to add a new case with an invalid description but with valid data in all other fields
6.9. Attempt to add a new case with an invalid date opened but with valid data in all other fields
6.10. Attempt to add a new case with an invalid case type but with valid data in all other fields
6.11. Attempt to modify an existing open case with an invalid description
6.12. Attempt to delete an existing case that is open

Task 7: Case Assignment Maintenance

- Display the cases' ids in a list.
- Allows the user to navigate between Cases in the list.
- Display a selected case's details (along with the client's full name and the full names of each investigator assigned) in a read-only format.
- If the user clicks on the **Assign Investigator** button and the selected case has a status of Open then
    - The **Modify Assignment** and **Remove Investigator** buttons are disabled.
    - A panel should appear (showing **Save Assignment** and **Cancel** buttons) allowing the user to enter new values for the Investigator (using a combo box), date (using a date picked) and hours.
    - If the user enters valid data for all fields and clicks on the **Save Assignment** button then a new assignment record is saved in the database.
    - If the user tries to assign an Investigator to the case that has already been assigned then the error message "This Investigator has already been assigned to this case" is displayed.
    - If the user clicks on the **Cancel** button then the panel should disappear and the **Modify Assignment** and **Remove Investigator** buttons enabled.
- If the user clicks on the **Assign Investigator** button and the selected Case has a status of Closed then the error message "Cannot assign investigators to closed cases" is displayed.
- If the user clicks on the **Modify Assignment** button and the selected Case has a status of Open then
    - The **Assign Investigator** and **Remove Investigator** buttons are disabled.
    - A panel should appear (showing **Update Investigator** and **Cancel** buttons) allowing the user to change the values for the assignment's number of hours and date (using a date picker).
    - If the user makes a valid change to the number of hours or date and clicks on the **Update Investigator** button then the Case Investigator record is updated in the database.
    - If the user clicks on the **Cancel** button then panel should disappear and the **Add Case** and **Delete Case** buttons enabled.
- If the user clicks on the **Modify Assignment** button and the selected Case has a status of Closed then the message "Cannot modify Investigators on a closed Case" is displayed.
- If the user clicks on the **Modify Assignment** button and the user is trying to set the assignment's hours back to zero then the message "Cannot reset an assignment's hours to zero" is displayed.
- If the user clicks on the **Remove Investigator** button and the selected Case has a status of Open then the user is asked to confirm the removal – if the user confirms the removal then the Case Investigator record is deleted.
- If the user clicks on the **Remove Investigator** button and the selected Case has a status of Closed then the error message "You may only remove Investigators from open cases" is displayed.
- If the user clicks on the **Return** button then focus is returned to the Main Menu.

Required tests:

7.1. Add a new assignment to an open case
7.2. Modify an existing assignment with valid data
7.3. Remove an existing assignment (that has a zero number of hours) from an open case
7.4. Return to main menu
7.5. Attempt to Assign a new Investigator to an Open Case with an invalid number of hours
7.6. Attempt to Assign an Investigator to an Open Case that has already been assigned
7.7. Attempt to modify an existing Investigator with an invalid hours
7.8. Attempt to add a new assignment to a closed case
7.9. Attempt to modify an assignment's number of hours back to zero
7.10. Attempt to delete an existing assignment that has a number of hours greater than zero

Marking

It is expected that the code for each of the program tasks that you finish does not contain compilation errors (i.e. no marks will be awarded if the code does not compile). Your application must update the database every time the user changes the data. You are also expected to handle exceptions by using MessageBoxes and "**try and catch**" blocks.

Requirement	Marks
Task 1: Main Menu (with testing documentation)	14
Task 2: Report "Invoices" (with testing documentation)	16
Task 3: Client Maintenance (with testing documentation)	10
Task 4: Investigator Maintenance (with testing documentation)	10
Task 5: Equipment Maintenance (with testing documentation)	6
Task 6: Case Maintenance (with testing documentation)	16
Task 7: Case Assignment Maintenance (with testing documentation)	18
Naming conventions and correct internal documentation	4
Checkpoints	6
Total:	100

Notes on marking

- Marks will be deducted for any task that is either not fully functional or not tested adequately.
- Check "Programming Standards for C Sharp Courses" at the end of this document. The standards for C# programming in this document MUST be followed. In particular this includes putting meaningful comments at the beginning of **each and every** method in the standard format as given in the document.

Do you want to do the best that you can do on this assignment and improve your grades?
You could:

- Talk it over with your lecturer
- Visit Te Puna Ako or Maia for learning advice and support
- Visit the Centre for Pacific Development and Support
- Contact the USU Advocate for independent advice
- For contact details and more information, go to www.usu.co.nz

Notes on exercises and assistance

Before approaching the lecturer for assistance with this assignment please ensure that you have completed **all** of the following exercises:

- Class 02 Database Exercise 1
- Class 04 Database Homework 1
- Class 04 Database Homework 2
- Class 06 Database Homework 3
- Class 06 Database Homework 4
- Class 08 Database Homework 5
- Class 08 Database Homework 6
- Class 08 Database Homework 7
- Class 08 Database Homework 8

Deliverables

Hard Copy

The hard copy is to be handed in by you individually at the beginning of class (5:30 p.m.) on Monday 13th April 2015 – 15 marks will be deducted if this is not done or late. The hard copy is to include:

- A standard cover sheet – available on Moodle.
- Testing documentation.
- Screen shots of all of your forms.

Soft Copy

The soft copy of the assignment is to be submitted in one ***.zip** file via Moodle and is to include:

- A soft copy of ALL files needed to compile and run your application from the **Visual Studio .NET 2010** environment used in the Department of Computing labs. Please note that you must submit your assignment in **Visual Studio .NET 2010**. 30 marks will be deducted if this is not done. There is nothing to stop you from using other versions of **Visual Studio .NET** when working at home **but** please note that it is **your** responsibility to convert your assignment into **Visual Studio .NET 2010** prior to submitting it (you are advised to test the conversion of your project on a regular basis).
- Testing documentation.
- Screen shots of all of your forms.

Demonstration

- You are required to demonstrate your program in class on Monday 20th April 2015. 10 marks will be deducted if you do not demonstrate your program.

User Interface

Always provide the users with clear instructions explaining what they should do. Areas used for input must be labelled to explain what input is required. Use **hints** or **tool tips** to explain interface features. The user must be prevented from entering values or taking actions that the program is unable to deal with. All input should be validated (use the MS Access database tables to see what is allowed in each field) and to see which fields are required); any errors found should be reported back to the user with an error message which clearly and politely explains how to correct the error. The user should be unable to proceed without correcting invalid input.

Assignment Delivery

Electronic submission of all of the necessary files is required for ALL assignments and must be submitted prior to 8:30am on the due date.

Assignments submitted after the due date and time without having received an extension through Special Assessment Circumstances (SAC) will be penalised according to the following:

- 10% of marks deducted if submitted within 24hrs of the deadline
- 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline
- 30% of marks deducted if submitted after 48hrs and up to 72hrs of the deadline
- No grade will be awarded for an assignment that is submitted later than 72hrs after the deadline.

Special Assessment Circumstances

A student, who due to circumstances beyond his or her control, misses a test, final exam or an assignment deadline or considers his or her performance in a test, final exam or an assignment to have been adversely affected, should complete the Special Assessment Circumstances (SAC) form available from Student Central.

Within any semester, a student may have only one SAC per course.

When requesting an SAC for an assignment, the SAC application form must be submitted (along with work completed to-date) within the time frame of the extension requested; i.e. if the Doctor's certificate is for one (1) day, then the SAC application form and work completed must be submitted within one (1) day.

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

- 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.

Progress Checks

You are expected to load your assignment onto Moodle on a regular basis so that the lecturer can monitor your progress. Failure to do so will result in marks being deducted. You will lose 2 marks for each progress check that you miss.

Progress Checks Schedule:

Progress Check Number	Progress Check Date
1	**23rd March 2015**
2	**30th March 2015**
3	**6th April 2015**

Programming Standards for C Sharp Courses

Internal Documentation

- Your code is such that other programmers can read it without struggling and your users are not left guessing as to what to do.
- Each class file (including form classes) will begin with comments explaining the purpose of the class, the author and the date written.
- Each method will start with comments that explain what the method does.
- Any code which does not have an obvious meaning or which uses a specialized technique is to be commented. Use blank lines and further comments to identify where parts of a task begin within a method.
- Code will use meaningful variable, class and method names. Components which have event handler code for any of their events must have meaningful names. Components which have properties assigned to in code must also have meaningful names. A naming convention that identifies the type of component involved is recommended. e.g. btnExit, txtStartDate, lblTotal

Layout

- Code will be laid out in the style of the example below, using indentation steps of 4 spaces. Blocks using { and } will use the layout shown here:

```
///<Summary> method : btnLeapYear_Click
///Check if a date falls in a leap year
///</Summary>
private void btnLeapYear_Click(object sender, System.EventArgs e)
{
    DateTime aDate = getDate();
    if ( DateTime.IsLeapYear(aDate.Year) )
    {
        label6.Text = aDate.Year.ToString() + " IS a leap year";
    }
    else
    {
        label6.Text = aDate.Year.ToString()
                        + " is NOT a leap year";
    }
}
```

- Parentheses and spaces will be used to make the meaning clear in arithmetic expressions and conditions:

```
sum = (n1 / n2) + n3;
```

not

```
sum = n1 / n2 + n3;
```

not

```
sum=n1/n2+n3;
```

- In general each method will perform a single simple task.
- In the final version of your project please delete all sections of code that have been ‘commented out’