Swinburne University of Technology
Faculty of Information and Communication Technologies

INF10016 / 60008 – Introduction to Programming in VB.Net
Semester 1, 2014

ASSIGNMENT 3

Must be done individually

Must be submitted via ESP by 8:30am Monday, 26th of May, 2014.
No extensions are available as this is an optional assignment.

Worth **10%** of your grade in this subject

Submission Requirements
You must submit all source code (the project folder), all executables and any other required files or folders in a **single** ZIP file via the **ESP submission system**.
https://esp.it.swin.edu.au

The name of the .ZIP file must be INF10016_#######_Ass3
where ######## represents your Student Id.

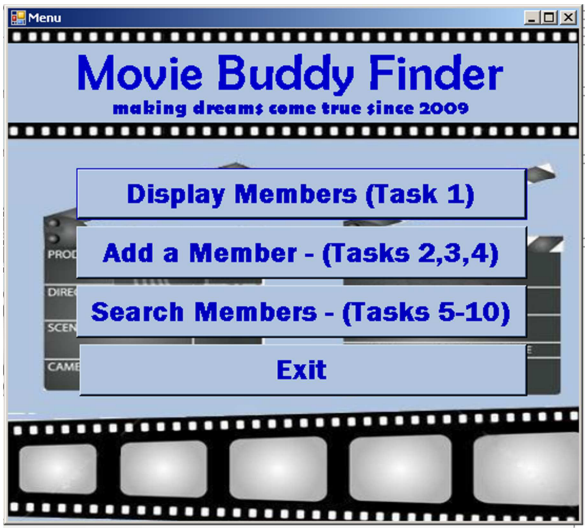
DO NOT USE A .RAR FILE FOR SUBMISSION

Testing
Please test your system fully.

Once you have zipped your files, copy and install your program on another computer (which does not contain your assignment). Check that all the required files can be unzipped and that your system then executes correctly.

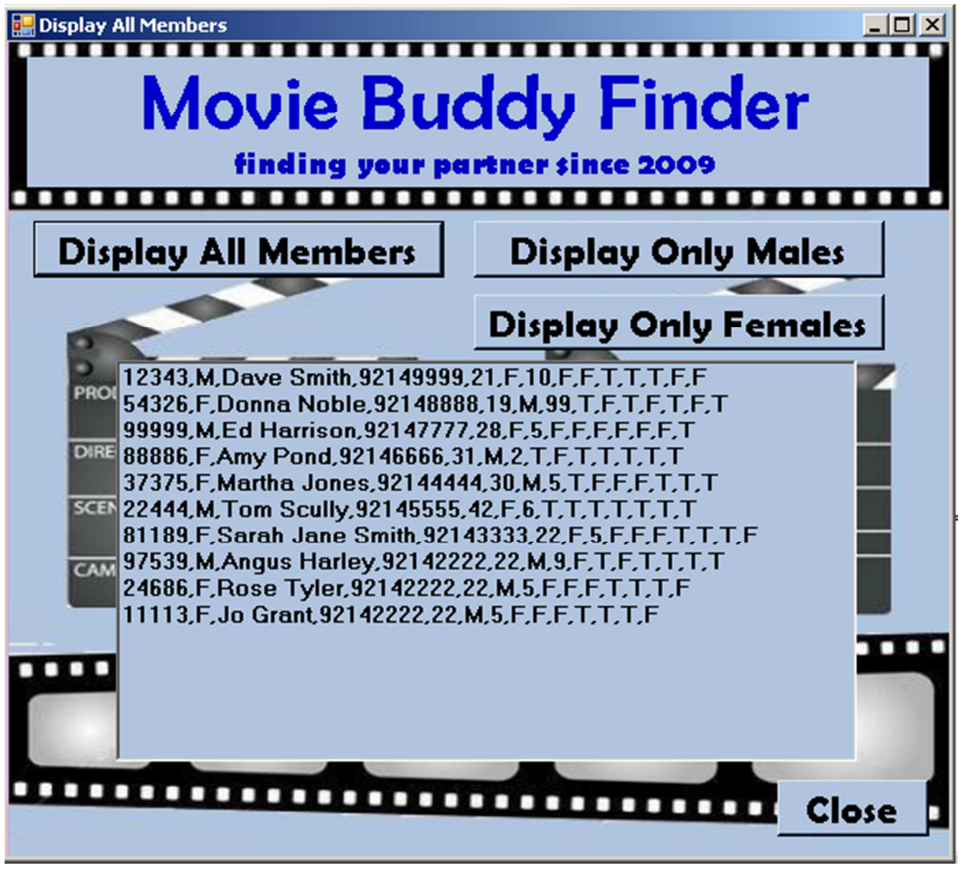
Test everything. **Twice**!!!!!!!!!

Create a Menu (colours, fonts etc may change) for the **Movie Buddy Finder** application



Task Number 1

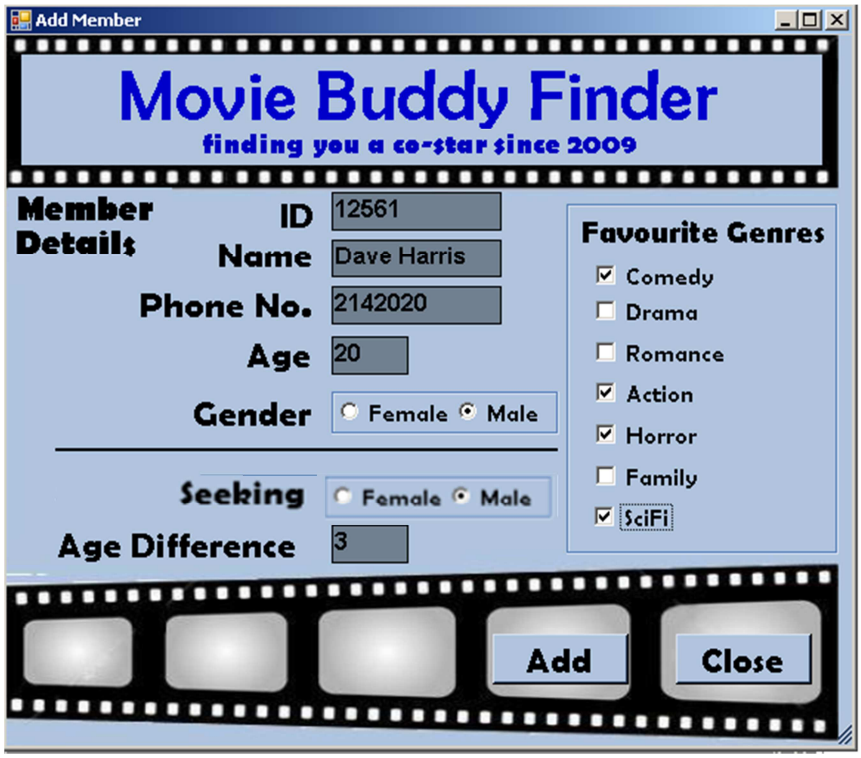
- Create a Display Members form as shown (colours, fonts etc may change)
- When the user clicks the Display All Members button
  - The application must display a file named "**C:\temp\members.txt**" (A sample file can be downloaded from Blackboard)
  - If the file does not exit then display an appropriate error message
  - Read each line in the file that the user selected and add it to a listbox
- When the user clicks the Display Only Males, do the same as above, but only display Males
- When the user clicks the Display Only Females, do the same as above, but only display Females



Task Number 2

Create an Add Member form similar to the one below (colours, fonts etc may change)

- The default radio buttons must be Female & Male
- The default value for all check boxes is checked, except for Drama, Romance, and Family checkboxes which should be unchecked (see image below).



Button Requirements:

- When Add Button is clicked:
  - If ID length is not = 5 then display an error and exit
  - If the Name length not in the range 2 to 30 then display an error and exit
  - If the Phone No is not numeric and the length is not between 8 and 10 characters in length then display an error and exit
  - The Age must numeric and in the range 18-99. If not display an error and exit
  - The Age Difference must numeric and in the range 0-99. If not display an error and exit
  - Append the data to a file named **"C:\temp\members.txt"**
  - Each piece of data will be separated by a comma.
    - Write F for Female or M for Male
    - Write T if Comedy is checked, otherwise write F
      - Similar for Drama, Romance, Action, Horror, Family and SciFi

    e.g. **12561,M,Dave Harris ,92142020,20,M,3,T,F,F,T,T,F,T**

Task Number 3
Modify how Task2 behaves
Before writing data to the file, ensure that **Member ID is unique** and does not already exist in the Student File
If the Member ID already exists, then display an appropriate error message

Task Number 4
Modify how Task2 and/or Task3 behaves
Before writing data to the file, ensure that Member ID check digit is valid
This must be done by writing a Method named **CalculateCheckDigit**().

- This method has one Parameter – a string which stores the 1st 4 digits of a member id
- This method returns an Integer – the check digit value

If the value returned by **CalculateCheckDigit()** does **not** match the 5th digit of the Member ID then the Member ID is **not** valid. Display an appropriate error message and exit the process.

Calculate the check digit using this process:

- Mulitply the 1st digit of the ID by 3 giving A.
- Mulitply the 3rd digit of the ID by 7 giving B
- Add A & B giving C
- Add C **plus** the 2nd digit of the ID **plus** the 4th digit of the ID giving D
- If the number of digits in D is greater than 1 then , add all the digits of D together and store the result into D
    - Repeat this until D contains a 1 digit number.

If D matches the 5th digit of the member ID, the member ID is valid.
If D does not match the 5th digit of the member ID, member ID is invalid

Examples

Student ID	Check Digit is
23423	valid
12343	valid
54326	valid
99999	valid
88886	valid
12345	invalid
54321	invalid
88888	invalid

Example Calculation:

23423	
2 * 3 = 6	(1st digit * 3 giving A)
4 * 7 = 28	(3rd digit * 7 giving B)
6 + 28 = 34	(Add A + B giving C)
34 + 3 + 2 = 39	(Add C + 2nd digit +4th digit giving D)
3 + 9 = 12	(Add digits of D giving D)
1 + 2 = 3	(Add digits of D giving D)
Check Digit is 3	

Task Number 5
Use **String handing techniques** to solve this task.
Read the members in the file named **"C:\temp\members.txt"**
For each member,

- display the initial and surname of each member
  - e.g. Fred Smith is displayed as F. Smith
- display the person's gender as Male or Female

F. Smith, M
S. Jones, F

Do **not** use the Split method (or similar) to solve this problem

Task Number 6
You must use the **Split method** and an **Array** to solve this task.
Read each member in the file named **"C:\temp\members.txt"**
For each member,

- Split the string that you have read into a 1-dim array
- Now display the Name, Gender and whether the member is seeking a male or female in a ListBox

e.g. Fred Smith is a Male seeking a Female
Sue Jones is a Female seeking a Male
Ted Black is Male seeking a either gender

Tasks 7 to 10 use a Class named Member

Create a class named Member
The class must contain these instance variables:

Variable Name	Data Type
Member ID	String
Name	String
Phone	String
Age	Integer
Gender	Char
SeeksGender	Char
SeeksAgeDifference	Integer
LikesComedy	Char
LikesDrama	Char
LikesRomance	Char
LikesAction	Char
LikesHorror	Char
LikesFamily	Char
LikesSciFi	Char

Tasks Number 7
Create an ArrayList named MemberList
Read all the member from the file named **"C:\temp\members.txt"**
For each member
Store the member's data into a **Member object**
Add the member to MemberList
(actually add a **reference** to the member object that was created)

Once all members have been added to MemberList do the following:

- Loop thru all the members stored in MemberList and display these details

Name	Phone	Age
Jeff Jones	92145555	24
Angela Loft	92143344	19
Toni Taylor	92144422	21

Task Number 8
Modify the Class named Member.
Add a method named **FavouriteCount**() to the Class that returns the number of Genres that the member has selected
e.g. If Fred has selected Comedy, Romance and Horror as his favourite genres, then FavouriteCount must return 3.

Add all the members in **members.txt** to an ArrayList (same as Task 7 above).

Then for each member in the ArrayList, pass the member to a sub procedure named DisplayMember()
The method header must be:
Sub DisplayMember(ByVal aMember As Member)
The method must use a message box to display the member's details in the following format:

Task Number 9

Add all the members in **members.txt** to an ArrayList (same as Task 7 above).

Ask the user to enter a Member ID.
If the member ID does **not** exist, then display an error message.
Otherwise, display the member's details using DisplayMember() (see Task 8)

Task Number 10
Add all the members in **members.txt** to an ArrayList (same as Task 7 above).

Ask the user to enter a Member ID.
If the member ID does not exist, then display an error message.

Search the ArrayList to find members who are a 'match' for the member ID
A match occurs when:

Two members must have the gender that each is looking for AND
The FavouriteCount() value each is the same +/- 2 AND
Two member's age ranges match

e.g.

Member Age	Member Age Difference	Other Member Age	Other Member Difference	Match
21	4	23	2	Yes
25	5	27	1	No
20	2	24	10	No
20	30	35	99	Yes

Use DisplayMember() to show each match.

Marking Scheme:

- The total marks available for this assignment is 50. This assignment is 10% of your overall marks.
- Each of the tasks above is worth 5 marks.
  - 5 marks will be awarded for each task that is completed and works perfectly or nearly perfectly

Comments:

- You must use the techniques specified for each task otherwise you will not score marks for that task.

- While the overall marks for the assignment are not large (10% for the entire assignment), these tasks align themselves closely to the final exam. This assignment forms a major part of your exam revision

- The file "C:\temp\members.txt" is used in many of the tasks.
  It **must** exist in this location. There is **no need** to use OpenFileDialog.