Topic 1B

Program Development Life Cycle

Topics

- What is Programming
- Program Development Life Cycle

Objectives:

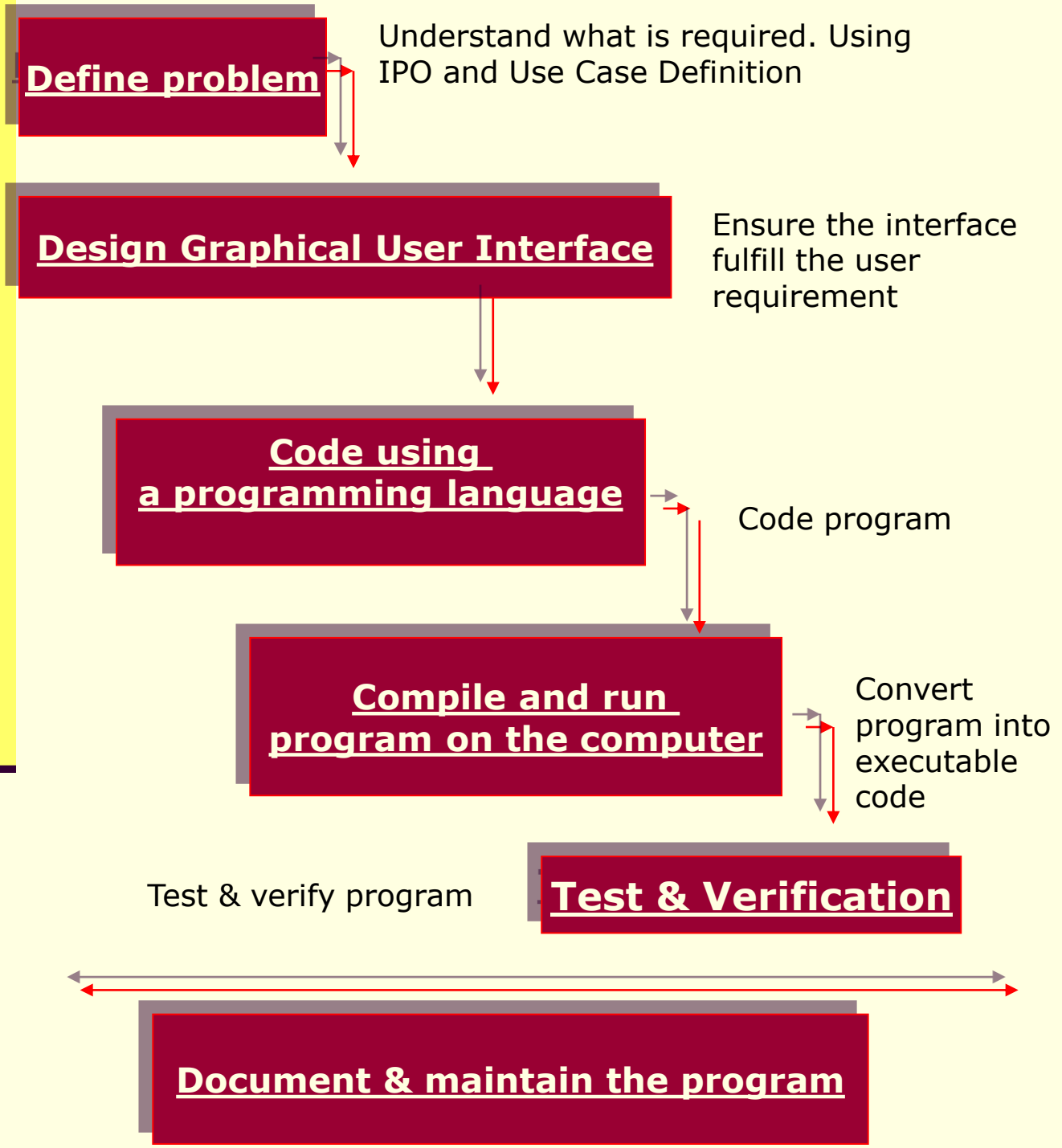
- Be able to describe the steps in program development process
- Be able to define the inputs/outputs and main processing tasks given a problem

Program Development Process

- What is programming?
  - Can be defined as the development of a solution to an identified problem.

- The phases in Program Development Life Cycle are:
  1. Define the problem
  2. Design User Graphical Interface
  3. Code the algorithm using a programming language
  4. Compile and run the program on the computer
  5. Test & verify program
  6. Document & maintain the program

Program Development Process



1. Define the problem

- **Understand what is required**
  - Generally, problem can be divided into 3 components
    1) Inputs
    2) Outputs
    3) Processing steps required to generate output
  - Example :

    Develop a program to read two numbers, add them together, and print their total

1. Define the problem

Develop a program to read two numbers, add them together, and print their total

IPO Table

Input	Processing	Output

1. Define the problem

- **Determine the inputs and outputs**
  - Look out for nouns & adjectives in the specifications
    - *Develop a program to read **two numbers**, add them together, and print their **total***

Input	Processing	Output
Number_1 Number_2		total

1. Define the problem

- **Determine the processing required**
  - Look out for verbs & adverbs used in the specifications
    - *Develop a program to **read** two numbers, **add** them together, and **print** their total*

Input	Processing	Output
Number_1 Number_2	Read Two numbers Add numbers Print total	total

1. Define the problem

- **Use Case Definition**
  - To clarify the requirement, we Use Case Definition
  - It is a sequence of actions a user will perform when using the program
  - It specifies each of these sequences of actions by describing what the user will do and how the program will respond to the user.

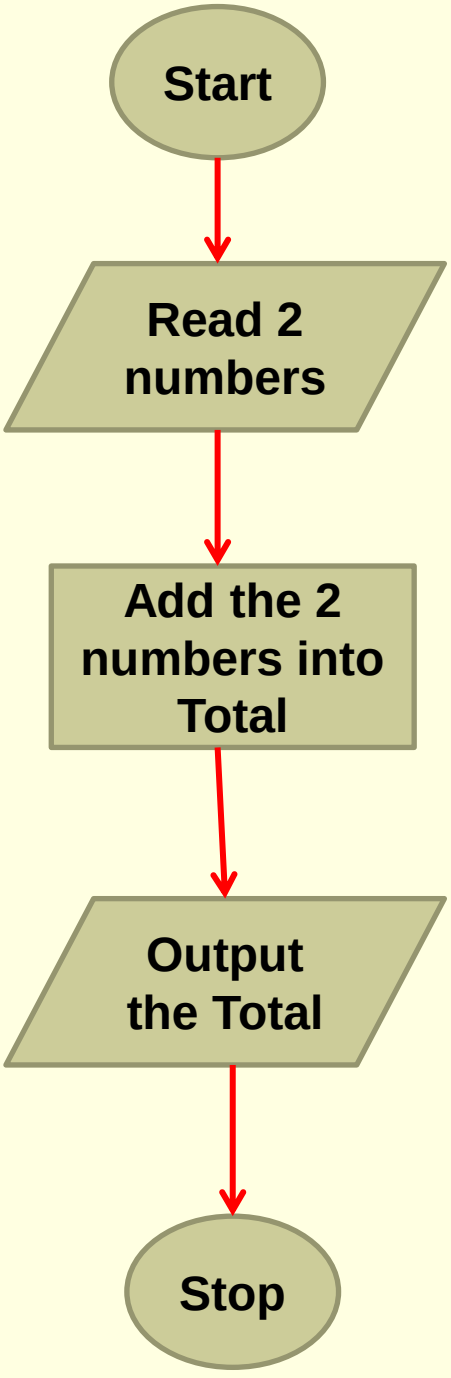
Example:

*Develop a program to **read** two numbers, **add** them together, and **print** their total*

Use Case Definition

1. User enters number 1 value
2. User enters number 2 value
3. User clicks total button
4. Program calculates total
5. Program displays total

Flow Chart



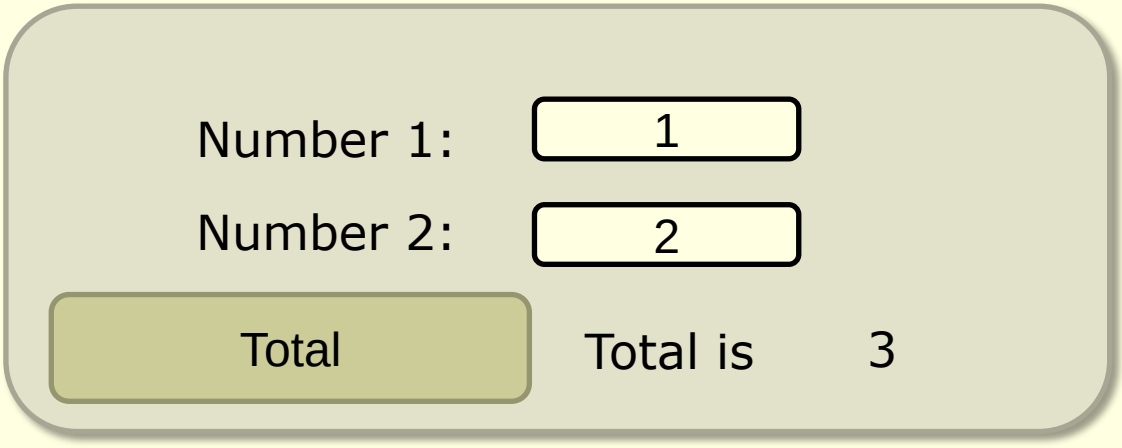
Pseudo code

Get 1st number.
Get 2nd number.
Add the 2 numbers into Total.
Display Total.

Ask for 1st number.
Input & validate 1st number.
Ask for 2nd number.
Input & validate 2nd number.
Add the 2 numbers into Total.
Display Total.

2. Design Graphical User Interface (GUI)

- Graphical User Interface consists of a window and contains a variety of objects
- Objects are form, buttons and labels
- Based on the Use Case Definition, design the GUI
- It defines how the user uses the application
  - *Develop a program to read **two numbers**, add them together, and print their **total***



GUI design

2. Principles of GUI design

1. Easy to use and follow by user. It must be user friendly.
2. GUI includes the windows, graphics and text shown on screen, as well as the methods used to interact with your program.
    - E.g keyboard, button
3. GUI should feel natural and normal
4. Consistent look's and feel
5. Objects in the interface MUST be arrange in the sequence in which they are used
6. Interface should be kept as simple as possible
7. Interface should be intuitive (No need for user guide!)

https://msdn.microsoft.com/en-us/library/aa468595.aspx

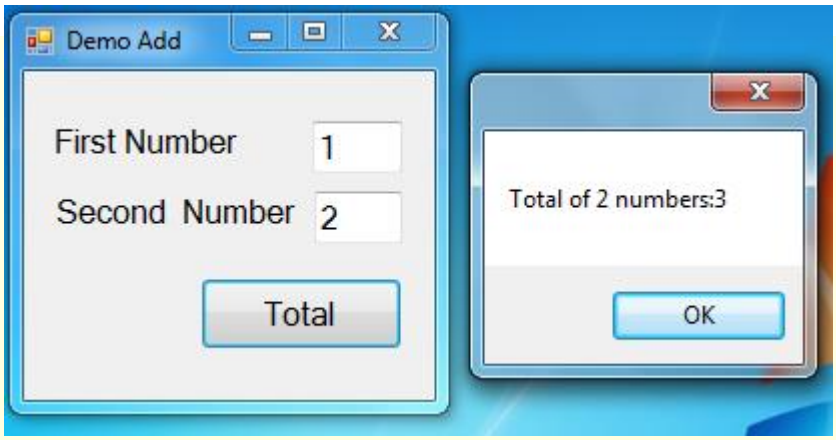
3. Code the algorithm

- After all the design considerations are met, you can start to code the program

You will learn C# in week 2

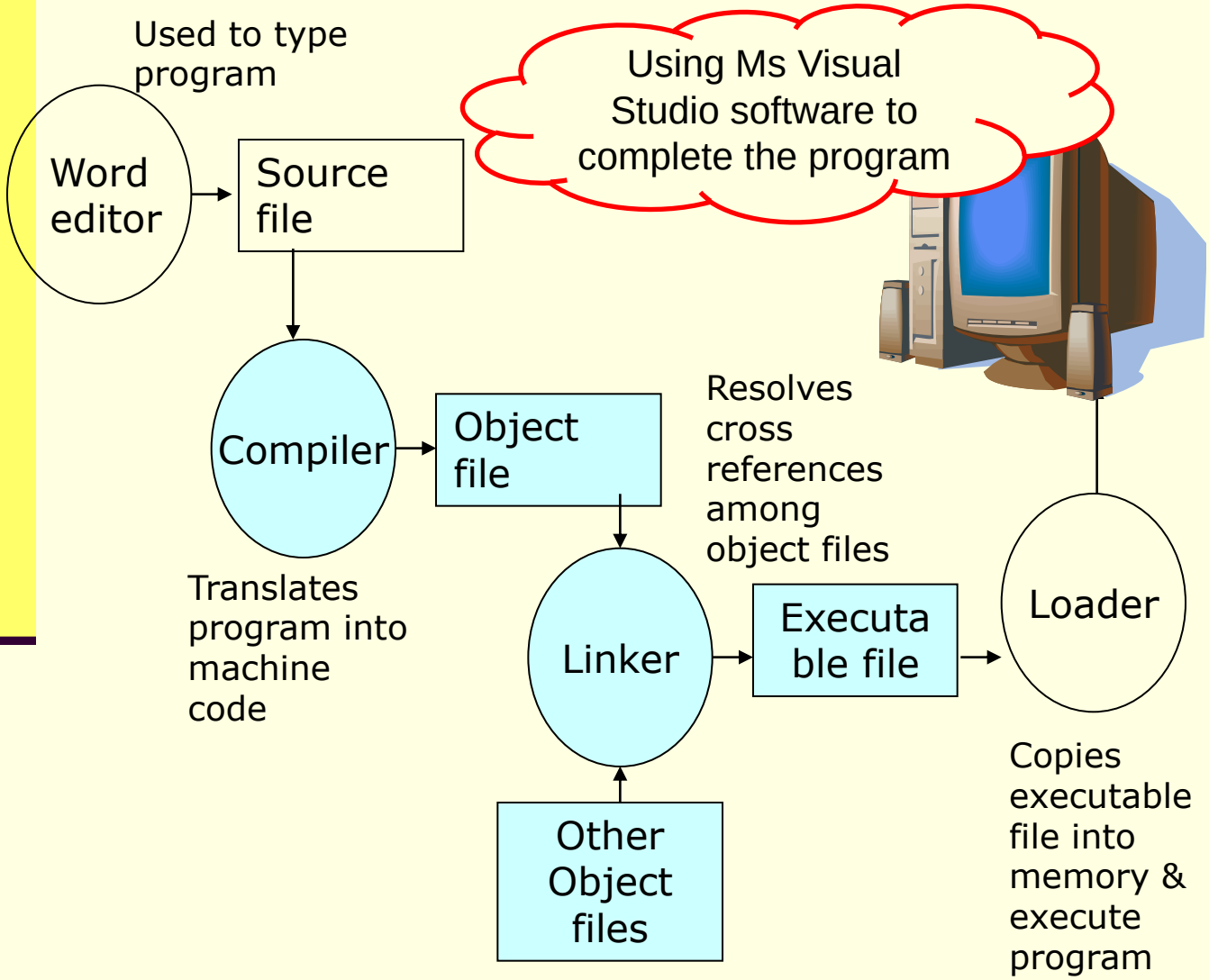
EXAMPLE : in C# code

```
private void btnAdd_Click(object sender, EventArgs e)
{
    int number1, number2, sum;

    // read in 2 numbers
    number1 = int.Parse(txtNum1.Text);
    number2 = int.Parse(txtNum2.Text);

    //calculate total
    sum = number1 + number2;

    MessageBox.Show("Total of 2 numbers:" + sum);
}
```



4.Compile and Run the program

- Uses a compiler to convert your program into executable code.



5. Test and verify program

- **Program testing** is to ensure its correctness
- **Program verification** is to ensure user requirements are met

Example of a test document :

Test Case	Test Description	Inputs	Expected Output	Actual Output	Remarks
1	Add 2 +ve numbers	2, 3	5		

6. Document and maintain your program

- Documentation is an on going process throughout the program development process

- Documentation includes
  - External documentation (problem definition, structure chart, solution algorithm, test data and results)
  - Internal documentation (which includes that coded into the program)

•Problem definition
•Structure chart
•Algorithm
•Test document
•Program (code)

- Program maintenance
  - Refers to changes which may be needed to the program throughout its life

Summary

The 6 steps in program
development life cycle

1. Define a problem using IPO and Use Case Definition
2. Design GUI
3. Code the algorithms in a program
4. Compile program
5. Test the program
6. Document and maintain

Practical 1- Blackboard

1. **Login to BlackBoard**
   using your student portal password

http://learn.nyp.edu.sg

2. Visit IT1753 Course

3. All lecture notes are available in the BlackBoard

4. Visit week 1 folder and download the lecture, practical demo and startup files

Practical 1
Computer Algorithms & Program Design

For each of the following questions, do the following:

- Define the problem by filling the IPO (Input Process Output) table
- Create the Use Case Definition
- Design the user interface (form design)
- Write the Pseudo code
- After completion, submit your work in MS WORD

Question 1 (example)

Design a program that prompts and reads the age of 2 different people, calculates the average age, and displays it.

Define IPO table

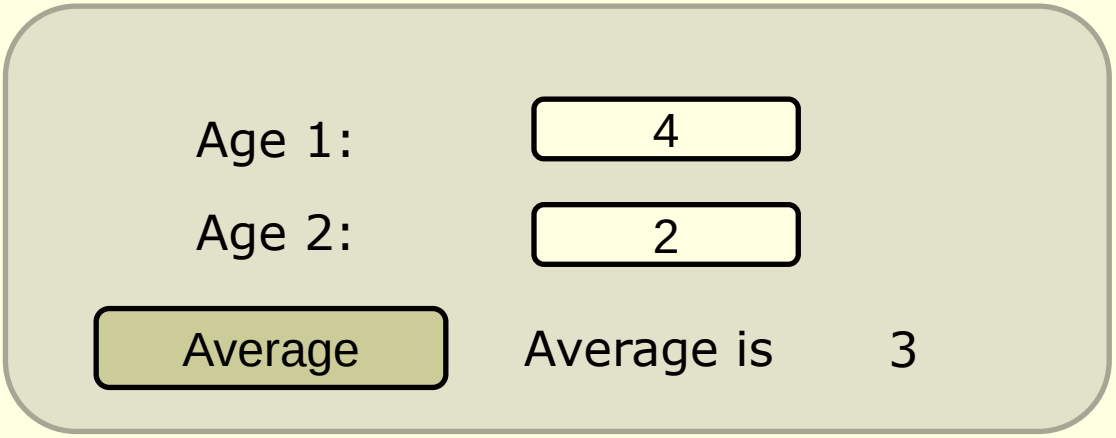
Inputs	Processing	Outputs
age1 **age2**	**Prompt for age** **Read in age1, age2** **Calculate Average** **Display Average**	**average**

Use Case Definition

1. User enters age 1 value
2. User enters age 2 value
3. User clicks Average button
4. Program displays Average

Practical 1
Computer Algorithms & Program Design

Design and Draw the Graphical User Interface on **paper**



Pseudo code

Read in AGE1.
Read in AGE2.
Calculate AVERAGE = (AGE1 + AGE2) /2
Display AVERAGE.

Study the above IPO table, use case definition, form design and pseudo code, and understand how the solution is derived.

Complete questions 2 to 5 and verify your answers with the tutor.

Please keep and bring them to class in Week 2 and 3. We will be coding the C# applications. You will see your paper solutions come ALIVE!

Practical 1
Computer Algorithms & Program Design

Q2. Design a program that prompts and reads a temperature value in degrees Fahrenheit, converts it to degrees Celsius, and display the converted value. The formula used for the conversion is:

Celsius = (Fahrenheit – 32) * 5/9

Q3. Design a program that prompts and reads the weight of 5 different pieces of luggage, calculates the total weight, and then displays it.

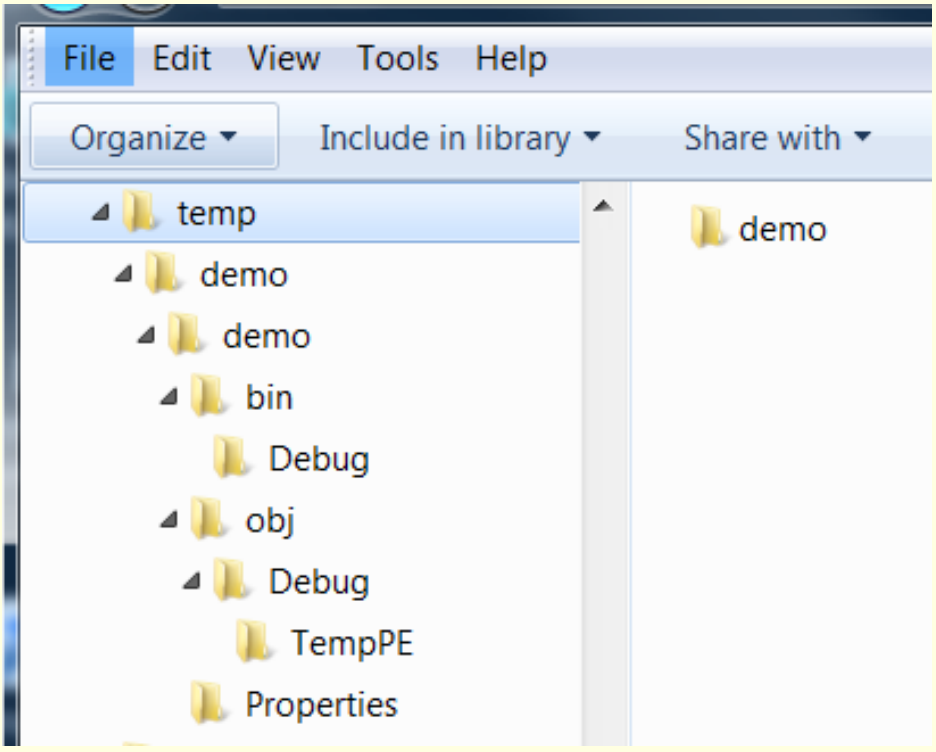
Q4. Design a program that prompts and reads in the price of an item, calculates the GST (which is 7% of price), and displays the GST value.

Q5. Design a program that prompts and reads in the height (in metres) and weight (in kg) of a person. It then calculates and displays the Body Mass Index (BMI) based on the following formula:

$$\textbf{BMI} = \frac{\text{weight (kg)}}{\text{height (m)} \times \text{height (m)}}$$

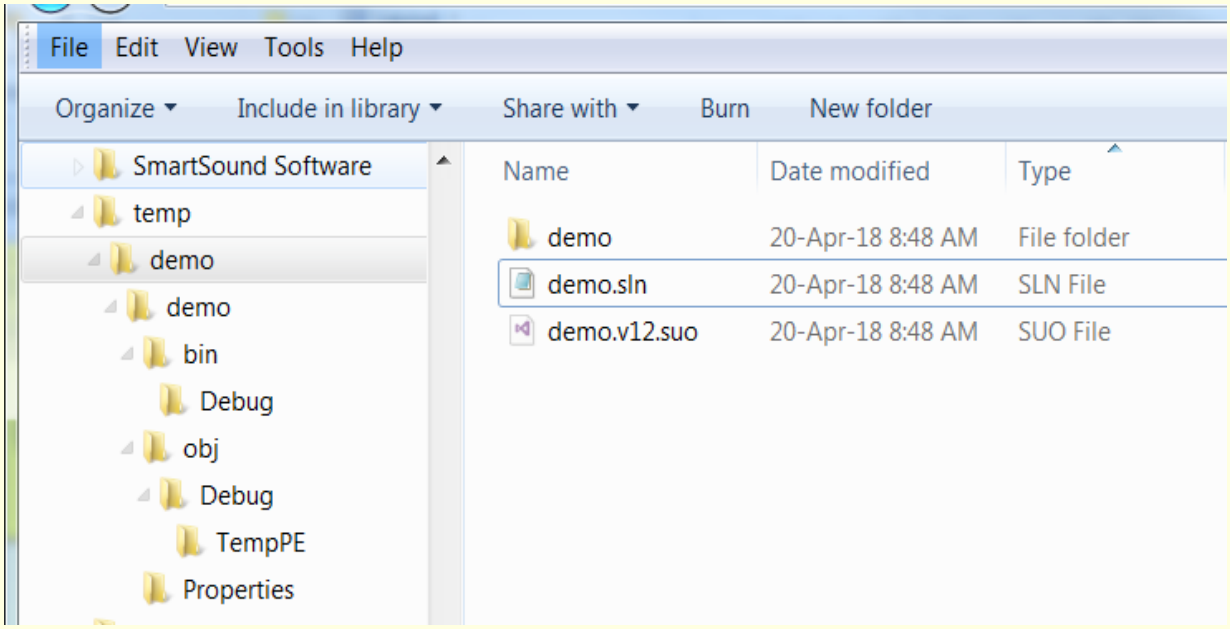
Visual Studio Project Files

- Optional – create the VS project based on slide 11 and 13.



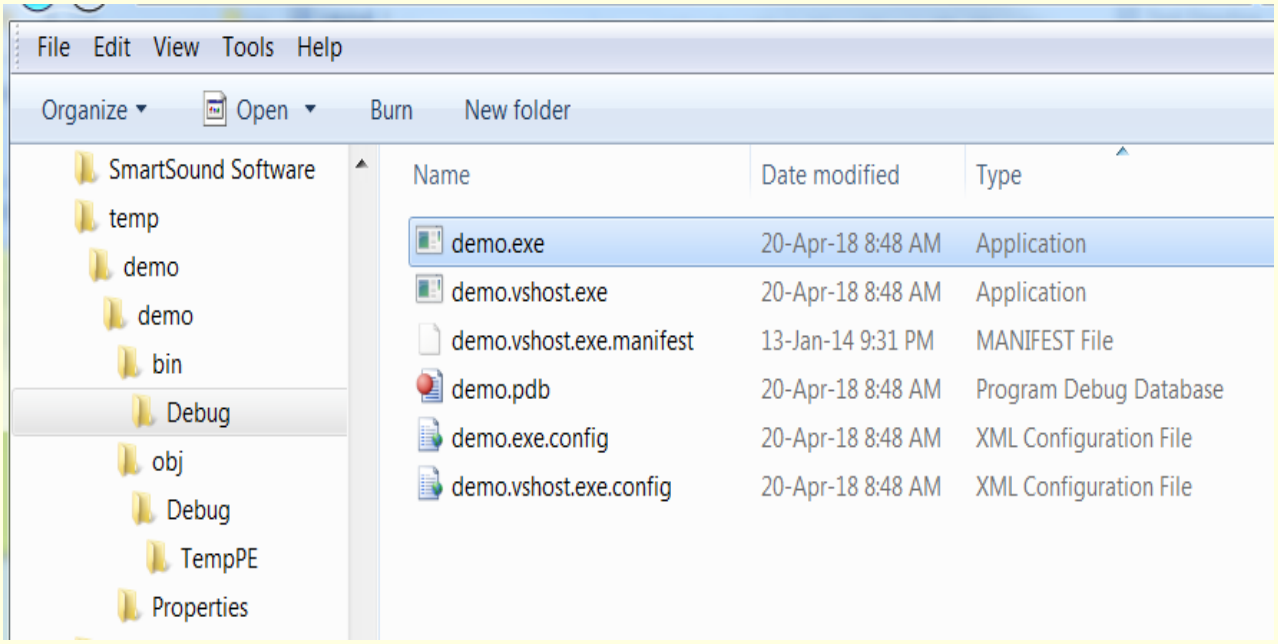
Which folder or files should you backup?

Visual Studio Project Files



- Which file to open in Visual Studio to open the project?

Visual Studio Project Files



- Which file to give to user to run the application?
- Which are the source file(s)?

End of Topic 1B

Program Development Life Cycle