

Simple program design

When we are given a problem to code, the first thing we should do is design the program BEFORE we even decide to write a line of code. This helps you to organise your thoughts and ensure that you have captured all the requirements of the problem.

The following steps are one suggested way of producing a solution:

1. Define the problem (with the customer if possible)
2. Write an IPO (Inputs, Processes, Outputs) table for the proposed solution
3. Write out Pseudo Code for the proposed solution. This is much easier to convert into code than you think.
4. Write the code
5. Test the project (Both Black Box and White Box testing)
6. Document the project in writing

Pseudo code

Pseudo code (literally means false code) is the way of writing the outline steps required in an algorithm but without writing any programming language code.

Pseudo code is structured English like way of representing the solution to a problem. It is considered a **first draft** because pseudo code eventually has to be translated in to a programming language. Although pseudo code is like English and has some precision to it, it does not have the very definite precision of a programming language. A computer cannot execute pseudo code. By using pseudo code it is easy to plan a program, allowing us to concentrate on the logic and not worrying about the rules of a specific language.

Pseudo code can be translated in to a number of programming languages including C#. You should not see anything that looks like computer code in pseudo code.

Let's assume we have been asked to write some code to get a password from the user. There can only be three attempts for the user before we display an authorised or unauthorised message.

Before we start to code, we should consider designing our code. For this module we will use Pseudo code **before** we start to write our program. The following shows one way of writing Pseudo code for the problem:

Set tries to 0

Do loop

Show number of attempts left

Ask user for password

Input password

If password is not equal to "secret" then

Output "Invalid password!"

Decrease the number of attempts left

End If

While attempts is not greater than 3 and password is not correct

If password correct then

Display "Welcome!"

Else

Display "Unauthorised user!"

End if

IPO Tables

Usually you are presented with a specification detailing the requirements of the programmed solution. It is from this specification that you:

- **identify the data to be *input* to the program**
- **determine the results to be *output* from the program**

The specification may detail the processes that are to be performed on the data, if not then further investigation into the problem will be required. From this the **Processes** are determined

Assume that a student takes three quizzes and the score for each quiz is the input. The output should be the average score of the three quizzes.

The *inputs* are:

- Quiz1 grade, Quiz2 grade, Quiz3 grade

The *outputs* are:

- Quiz Average

The *processes*, in outline, are:

- Calculate the average.

Inputs	Processes	Outputs
Quiz1 grade Quiz2 grade Quiz3 grade	Get inputs Sum the quiz grades Calculate average Display average	Average

Pseudo Code for the quiz problem

Begin

```
Get Quiz 1 grade
Get Quiz 2grade
Get Quiz 3 grade
Sum= Quiz1grade + Quiz2grade +Quiz3grade
Average = Sum/3
Display Quiz Average
```

End