

Professional Practice – Day 2 - Subprograms

Program Structure:

Our programs should be structured in a way that is easy to find everything, use the following order:

- Header
- Libraries
- Class Line
 - Global Variables
 - main subprogram containing program logic and calls to subprograms
 - Subprograms

Whitespace:

Previously we mentioned breaking up our code into blocks, since then we also discussed subprograms and selection statements which are themselves code blocks. With a selection statement, the overall statement itself is a block and requires a comment describing the overall purpose, however inside each option in the if statement there may be more code blocks that may also need a comment.

With subprogram we must remember two key places for whitespace in addition to our old style, before and after the definition of the subprogram as well as indentation of **ALL** code within the subprogram.

Documentation:

Similar to documenting selection statements a subprogram is a code block itself and needs to be documented to give an overview of its purpose, although in a unique way described below. However, the logic inside the subprogram is also comprised of one or more code blocks that each require documentation as normal.

To document the subprogram as a whole we will give each subprogram its own header that goes directly above the subprogram and is made up of three sections. We call this header the **PrePost Conditions**.

How:

Every subprogram requires three pieces of information directly above it:

- **Pre:** What are the parameters and assumed restrictions of this data. Write **None** if there are no parameters
- **Post:** What is the returning data and type. Write **None** if there is no return value
- **Description:** A brief description of what the purpose of the subprogram is. Sometimes some complications like equations may be stated to clarify the logic

Examples:

```
//Pre: length and width are a valid numbers greater than 0
//Post: Return the resulting area as a double
//Description: Calculate the area of a rectangle given a specific length and width
private double CalcRecArea(double length, double width)
{
    double area;
    area = length * width;
    return area;
}

//Pre: None
//Post: None
//Description: Output Stewie Griffin's mailing address to the screen
private void OutputStewiesMailingAddress( )
{
    string output = "Name: Stewie Griffin\n31 Spooner St.\nQuahog, RI\n43954";
    MessageBox.Show(output );
}
```