Code Documentation
Important!

- It’s for you
- It’s for others
The Function Description

- Describes what the function does for the user
- What resources are needed
- It should not need to state *how* this is done, just *what* is done
Function Side Effects

- A side effect is anything that is a direct consequent of the execution of a function that is not a direct mapping of inputs to outputs (return values)
Pre and Post Conditions

- **Preconditions**
  - Preconditions state what must be true of the inputs to the function before a function is executed in order to guarantee the postconditions.

- **Postconditions**
  - Postconditions state what will be true after successful execution of a function including all side effects given that the preconditions are met.
Pre and Post Notes

1. You may think of pre- and postconditions as setting up a contract between you (the programmer) and the user of your function.

2. It is perfectly acceptable that there can be no preconditions on a function.
Pre and Post Notes

3. Preconditions should never contain obvious, irrelevant, or redundant statements.

4. Most often you will find that preconditions will be a statement of the range of values a parameter may take on
Pre and Post Notes

5. When stating postconditions, be sure to include all pertinent information. Beware of reference parameters and other side effects.

6. Don’t state what is not a result of the function being executed.
Pre and Post Notes

7. I CANNOT imagine “none” being a valid postcondition. If the function had no postcondition, then it’s pretty useless.

8. All these comments should be placed just before the prototype of the function in question.
Examples

// Description: The greetings() function will output a message to the
// screen greeting the user.
// Pre: None
// Post: Message output to the screen.
void greetings ();

// The cyl_vol() function will calculate and return the volume of the
// right circular cylinder with base radius rad and height ht.
// Pre: Both parameters, rad and ht, must be positive values.
// Post: Volume of cylinder is returned.
float cyl_vol (const float rad, const float ht);

// The swap() function will swap the values of the arguments sent to
// it.
// Pre: none
// Post: The values of the arguments sent will be swapped back in the
// calling function.
void swap (float & val1, float & val2);
Examples

// Description: The greetings() function will output a message to the
// screen greeting the user.
// Pre: None
// Post: Message output to the screen.
void greetings();

// The cyl_vol() function will calculate and return the volume of the
// right circular cylinder with base radius rad and height ht.
// Pre: Both parameters, rad and ht, must be positive values.
// Post: Volume of cylinder is returned.
float cyl_vol(const float rad, const float ht);

// The swap() function will swap the values of the arguments sent to
// it.
// Pre: none
// Post: The values of the arguments sent will be swapped back in the
// calling function.
void swap(float & val1, float & val2);
Examples

// Description: The greetings() function will output a message to the
// screen greeting the user.
// Pre: None
// Post: Message output to the screen.
void greetings();

// The cyl_vol() function will calculate and return the volume of the
// right circular cylinder with base radius rad and height ht.
// Pre: Both parameters, rad and ht, must be positive values.
// Post: Volume of cylinder is returned.
float cyl_vol(const float rad, const float ht);

// The swap() function will swap the values of the arguments sent to
// it.
// Pre: none
// Post: The values of the arguments sent will be swapped back in the
// calling function.
void swap(float & val1, float & val2);
Examples

// Description: The greetings() function will output a message to the
// screen greeting the user.
// Pre: None
// Post: Message output to the screen.
void greetings ()

// The cyl_vol() function will calculate and return the volume of the
// right circular cylinder with base radius rad and height ht.
// Pre: Both parameters, rad and ht, must be positive values.
// Post: Volume of cylinder is returned.
float cyl_vol (const float rad, const float ht);

// The swap() function will swap the values of the arguments sent to
// it.
// Pre: none
// Post: The values of the arguments sent will be swapped back in the
// calling function.
void swap (float & val1, float & val2);
End of Session