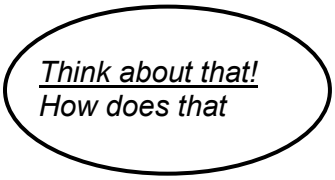
Commenting Guided Notes

Instructions:

Fill in the missing blanks in the notes as you watch the course videos for this learning module.

Code is written once and read many times .

The computing scientist's main challenge is to not get confused by the complexities of his own making.



- commenting is an internal documentation mechanism
- Well written code needs fewer comments.
- Poor commenting is a waste of time and sometimes harmful.

Commenting: The good, the bad, and the ugly.

- If good code is self documenting, should you never comment your code? No
- repeating the code is an example of a bad comment
- Avoid both bad and ugly comments.

Why are names important? Names are used everywhere in code. Names should:

- follow agreed style guidelines
- be informative (match the item's purpose and behavior)
- be concise
- be memorable and pronounceable
- be consistent.

Using comments to help.

You can place markers in the code as you develop.

Describe the code's intent. (the why) Not the how.

Comment Functions, Global variables, and code that is truly complicated.

Maintaining Comments

Comments need to be maintained:

- commenting styles can assist with maintenance .
- Comments need to be included in formal reviews .

- Don't use comments you don't want others to see.
- Document surprises not obvious in the code.

Summary

Use Comments to help the reader.

Summarize your notes in 2-4 sentences. Identify the key points and the main take-away message.

Commenting is something that should be added to code in a manner that is appropriately helpful to the reader. Comments should not restate the code itself, be inappropriate, or be over the top. Instead, we should comment to explain the code's purpose- especially in complex or surprising portions.