Good Programming Practice

1) **Comment and Documentation**
   Commenting is useful, it allow IDE and other tools to utilize them in different way for example when you call out some function they will have a description of the function where you've commented. Commenting also make the program better to edit improve or find out the problem. By documenting programmer will know what the program uses are without having to remember it in a pile of coding.

2) **Consistently Indentation**
   When writing a code we should be best to consistently indent the code, it is a good practice to make the coding logic and style clear. If you are part of a team or if you are contributing code to a project, you should follow the existing style that is being used in that project. Indentation make the code clear and easy to read.

3) **Avoid Obvious Comment**

   Avoid obvious comment, overdone comment could be redundant, the usage of the code is simply obvious to know its functionality then there will be no need of commenting in every single line of code, instead use commenting on the functionality of the whole function.

4) **Code Grouping**

   Grouping code is the best way to make code more tidy, sometime we only uses a few line of code for certain task and it is best to group them and space it from other code, it makes the code looks tidier and easy to read.

5) **Avoid deep level**

   It is best not to make code hard to read by adding a lot of level in the code, such as a lot of if else cases inside each other.

6) **Naming**

   Naming should be a good practice in programming, naming a function and class would give a clear functionality and readability to the writer and also anyone who going to access the code.

7) **Coding Line's Length**

   It is good to limit the line length to a readable length, it is hard to read and diagnose or revise code when there are code which have a really long line

8) **Unit Testing**

   Unit testing is a good practice to diagnose the error, when there is a big project with different people working on different part unit testing will provide a good idea of

coding error when combine or logically mistake. During working on the task I often find out unit testing shows a lot of result that I have not expected and able to find out the mistake of my logic, although the compile is successful but a small mistake have change the outcome. Unit test is a good way of testing before release or use of a program.

9) UML Diagram and UML Sequence

UML Diagram and UML Sequence enable me to have a clear view of logic before coding. It is useful during working of big project and complicated programming, it will help to clear things up.