1. Encapsulation used in OOP designs is the principle of hiding class information of an object and prevent from unrestricted public access. Encapsulation in OOP enables the implementation of coupling by enabling other classes to associate themselves with the dataset that the encapsulated object provides via inheritance. Encapsulation in OOP also enables the implementation of cohesion as developers are encouraged to bear the single-purposeness of class methods in mind to ensure reusability of the methods.
2. i. Use naming conventions. Readable variable names allow other developers to track and troubleshoot points of interest more easily.

ii. Use indentation. Appropriate whitespace de-clutters software code and make program scopes much more readable.

iii. Use comments in code. Comments allow developers to understand a particular code section more easily and avoid making unnecessary side effects.

iv. Use adaptive variables. Developers will be able to make policy changes more efficiently using adaptive variables instead of static values effortlessly.

1. i) -Make changes for requirements that have been developed inaccurately.

-Make changes to procedures that produces wrong results.

ii) -Make changes to tuition policy changes, such as fee amount and timetable schedules.

-Make changes to external API usages in the program according to latest releases.

iii) -Make changes to code sections that do not have error handling.

-Make changes to coding conventions that have been used inappropriately in the code.