The definition of abstraction in program language is to hide actually workflow details from the user. Instead, the program just only shows to the user essential information only. It is because users do not need to know how the program run. That means the program just need to show what choices are available to users, and to hide how these choices work.

The benefit of abstraction is look simple and easy. Users just need to choose what service they need to make the human world life more convenient. It is practical. And it is user friendly because not everyone can understand program languages in the human world. So, it is not necessary to show the users the program working process.

For example, when we slice down our mobile screen, it will show a control panel with several buttons to let user to tap on or off different functions such as if the user tap the Wi-Fi button, the Wi-Fi function will be on, and if the user tap again the Wi-Fi button, the Wi-Fi function will be off. It is a very simple way for human to use our mobile phone.

The below is an example of my code:

using RandomQuestionGenerator;

/\*omit the middle part\*/

var randomQuestion = PromptQuestion.\_randomQuesionsPrompt();

This function can show a random question when the user press “1”. When the program gets a user input “1” to write journal, it will refer to the RandomQuestionGenerator.cs file, there are several questions were set in a list. And it will randomly pick one immediately. After it choose a random question from the list, the random question will go back to the program.cs file show to the user right after the user press “1” to choose write journal.