

The four principles that our original program attempted to break were: KISS, DRY, clean code, and documentation. Our overall program is a fortune teller that draws three tarot cards and tells the user their meaning. We broke the clean code principle by using bad variable and function names. For example, the function “myFunction” works to shuffle the deck of tarot cards so that the player receives a random card instead of the same one every time. Because of this, a better name for the function would be something such as `shuffle()`. Additionally variables were given names such as “usuallyTrue”, “myDict”, “num”, and “a”. All of these variables and functions should have been given better names that fit their purpose.

Not helping with the readability is the documentation. Comments were placed haphazardly, and described what the code did in literal terms. For example, the code that was used to choose a new random card, was commented as “adds a random number to num and finds the modulus of the dictionary’s length”. A better comment would be something along the lines of “chooses a new card”.

The program violated the DRY principle by repeating code when the play drew a card. While a `drewCard()` function was added, it only handled a small portion of the logic. Lines 84-93 and 95-104 were basically identical, and lines 72-81 were very similar. Finally the program violated the KISS principle by shuffling the deck constantly in the background instead of generating a random number when it was time to draw. Making the program asynchronous was far more complicated than the alternative, and unnecessarily wasted computational power.