|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement** | | **Name** | **Location** | | **Notes** |
| **File** | **Line** |
| List | | *word\_list* | game.py | 12 | .splitlines() method returns an array |
| Tuple | | *self.\_\_secret\_word* | game.py | 11 |  |
| Set | | *self.guesses* |  |  |  |
| Dictionary | | *feedback* | game.py | 51 |  |
| Iteration | | --- | game.py | 40 | for loop used to compare words, one char at a time |
| Conditional | | --- | game.py | 41 | if statement drives comparison behavior |
| Try/Else | | --- | main.py | 13 | Try to read in data file. If exception, then exit |
| Input/Output File | | *word\_list.txt* | main.py | 14 | Read in text file containing all possible 4-letter words |
| User Defined Function | | *print\_feedback* | main.py | 22 | Translate *feedback* dictionary into meaningful text |
| User Defined Class | | *Game* | game.py | 7 | Game class contains secret word and game methods |
|  | init | *\_\_init\_\_* | game.py | 9 | Initializes secret word randomly from *word\_list* parameter |
|  | Private Attribute | *self.\_\_secret\_word* | game.py | 11 |  |
|  | Public Attribute #1 | *self.guesses* | game.py | 12 |  |
|  | Public Attribute #2 | *self.ongoing* | game.py |  |  |
|  | Private Method | *\_\_print\_win* | game.py |  |  |
|  | Public Method #1 | *validate* | game.py |  |  |
|  | Public Method #2 | *evaluate* | game.py |  |  |
|  | str() or repr() | *\_\_str\_\_* | game.py |  |  |
|  | Magic Class Method | *\_\_bool\_\_* | game.py |  | Used to determine if game was won or not |
| Unit Test #1 | |  |  |  |  |
| Unit Test #2 | |  |  |  |  |