To get a passing grade you'll need to submit a project that meets the following requirements. If your project receives a "Needs Work" rating on any of these tasks, then you'll need to edit your project and re-submit it until it "Meets" or "Exceeds" each task requirement.

	Needs Work	Meets Expectations	Exceeds Expectations
Script execution	• The script crashed early due to an uncaught exception • Script execution logic such as instance creation, method calls, or print statements were outside of `if __name__ == '__main__':` which cause code to automatically execute, if `app.py` is imported into another script	• The script should not crash due to uncaught exceptions. Raised exceptions should be handled appropriately so the program can continue or exit without a crash • Instance creation, method calls, print statements, or any calculated execution logic should be wrapped inside a Dunder Main statement for the entry point script of `app.py`	• N/A
Character initialization	Was missing one or more of the following: • An initializer method, `def __init__` • A parameter on `def __init__` that allows a single string character to be passed in and stored in an instance attribute	• The initializer method `def __init__` was implemented • A parameter was defined on `def __init__` to allow a single string character argument to be passed in and stored as an instance attribute on the Character object	• N/A
Phrase initialization	Was missing one or more of the following: • An initializer method, `def __init__` • A parameter on `def __init__` that allows a phrase to be passed in and stored in an instance attribute • The phrase was not composed of a collection of Character object instances	• The initializer method `def __init__` was implemented • A parameter was defined on `def __init__` to allow a phrase argument to be passed in and stored as an instance attribute • The phrase is composed of a *collection* of Character object instances	• N/A
Game initialization	One or more of the following occurred: • The initializer method `def __init__` was not implemented • `__init__` did not accept a `phrases` parameter that was stored as an instance attribute • The players turns/lives are not being stored on the game class • The active phrase was not set as an instance attribute on creation of the Game class	• Includes an initializer that will set a `phrases` instance attribute to a List of five `Phrase` objects • Initializes an instance attribute to store and track the players turns/lives which should start at `5` • Initializes an instance attribute to store the "active phrase", or the phrase the will be guessed by the player	• N/A
Entry Point app.py	Was missing one or more of the following: • The entry point file was either not named `app.py` • Running `app.py` did not create a Game instance and call the instance method to start the game loop.	• The entry point script was named properly as: `app.py` • Running `app.py` creates a Game instance and calls an instance method that handles starts the game.	• N/A

	Needs Work	Meets Expectations	Exceeds Expectations
Gameplay	Missing one or more of the following: • Prior to the player's first attempt at guessing the phrase did not output to the screen hidden, showing only _ placeholders for each character • The did not use strictly Object-Oriented approaches: Instance creation and method calls for the entire length of one or multiple plays of the game.	Prior to the player's first attempt at guessing: • The phrase should be outputted to the screen showing only _character placeholders for each character with noticeable spacing between each character. • One or more plays of a game used strictly Object-Oriented approaches meaning Instance creation and method calls for the entire length of one or multiple plays of the game.	• N/A
Correct character guess	Missing one or more of the following: • All occurrences of the character in the phrase were NOT shown when the player made the correct guess. • The remaining unguessed characters were also revealed instead of staying hidden as _ characters.	When a correct character guess is made by the player: • The phrase is updated on the screen so that all occurrences of the correctly guessed character in the phrase are shown to the player • The remaining unguessed characters are still hidden as _characters	• Validate the player's guess by ensuring the guess is 1 character in length and that it is only letter character: a through z *(uppercase or lowercase)* • Any errors/exceptions should be handled

	Needs Work	Meets Expectations	Exceeds Expectations
Incorrect character guess	One or more of the following occurred: • Guessing wrong did NOT reduce the player's remaining lives/turns by 1 • The player was NOT shown how many turns/lives they have remaining after guessing incorrectly	When an incorrectly guessed character is made by the player: • The player loses a turn/life up to 5 turns/lives • The player is shown how many turns/lives are remaining	• Validate the player's guess by ensuring the guess is 1 character in length and that it is only letters: a through z *(uppercase or lowercase)* • Any errors/exceptions should be handled **NOTE**: It is entirely up to you if you want to remove a turn/life for these types of incorrect guesses.
Player wins	One or more of the following occurred: • The player was NOT shown a winning message • The player received additional prompts for guesses after they won	When a player guesses ALL characters in the phrase: • The player was shown a winning message • No more prompts for guesses occur	After winning, the player was prompted to play again: • if they agree a new game instance should be created OR the current game instance should have its attributes reset to a NEW game state • if they disagree, the game should end with a message

	Needs Work	**Meets Expectations**	**Exceeds Expectations**
Player loss	One or more of the following occurred: • The player was NOT displayed a losing message • The player receives more prompts for guesses after the game is over	When a player runs out of turns/lives before guessing all the of the phrase: • The player was shown a: "Game over" losing the message • No more prompts for a guess should occur after the game ends	After losing, the player was prompted to play again: • if they agree a new game instance should be created OR the current game instance should have its attributes reset to a NEW game state • if they disagree, the game should end with a message