Hangman Treasure Hunt

**Problem**: You are a treasure hunter. You have been searching for the long lost treasure of Jackson Kennedy. You’ve fought every obstacle in your path, and now there is just one thing standing between you and the treasure: a computer. You walk closer to the computer screen and you see what looks like a set of instructions. It says: “Congrats! You’ve reached the treasure. Now, all that stands between you and riches is a game of hangman. Two words, both five letters long. There is a blind cyclops 6 steps to your left. Every time you guess a letter wrong this computer will make a loud noise and the cyclops will take a step towards you. You don’t want to know what happens when he reaches you. Enjoy!”

Program Requirements

1. Name your program Lastname\_FinalProject
2. On the top of your code put the following in a comment:
   1. Program Name:
   2. Programmer:
   3. Email:
   4. Purpose:
3. Add comments as appropriate.
4. Create a .txt document and save it in the same directory as your program. In the document write “Hello World” (without the quotes). This will be the answer to the hangman game. Name it whatever you want.
5. Lists needed:
   1. Solution: List with each letter of the solution.
   2. CorrectGuess: List with the user’s correct guesses (arranged correctly)
   3. IncorrectGuess: List with the user’s incorrect guesses.
6. Functions needed:
   1. readFile: Read the file with the solution in it and creates a list with each letter in it. Make sure to catch exceptions.
   2. checkInput: Checks if the user input is in the solution. If yes, tells the index where.
   3. correctInput: Adds the user’s correct input to the correct index of CorrectGuess.
   4. incorrectInput: Adds the user’s incorrect guess to a list and brings the cyclops one step closer.
   5. displayProgress: displays CorrectGuess (with periods in place for letters that haven’t been guessed), IncorrectGuess, and how far away the cyclops is.
7. Main Function:
   1. Tell the story, give background on the situation. You may use the same story as above or come up with your own (with the same fundamental details).
   2. Call readFile. Ask for the user’s guess, then call checkInput.
   3. If their guess is in the solution, call correctInput. If not, call incorrectInput.
   4. After each guess, display the progress.
   5. If the player wins, display a story about what the treasure is.
   6. If the player loses, display a story about what happens when the cyclops reaches you. Ask if they want to try again.