Hints to help building Guess The Movie game

Game play hints

In English, the top 5 frequency of letters is `e` `t` `a` `o` `i`. It can help you play this game after you finish it.

It's an important study in Cryptanalysis. More info about this, please read Letter frequency from Wikipedia.

Use classes

This program will have more code than all of the exercises we've previously covered, so it's a good idea to divide your code into classes instead of writing everything in 1 class

A simple design would be to have at least one more class called `Game` that will include methods responsible for handling a single guess or displaying the hidden movie title etc.

Then have another class that contains the main method and controls the logic of reading the user's input and calling the methods in the `Game` class

Build it step by step

Don't rush into building the entire game at once, start small, for example:

1. Write some code that will simply read the movie list and display the whole list.
2. Then add to your code to randomly pick one movie and display that title only.
3. Then convert its letters to underscores (_) and display that instead, and so on.

4. Once you've got that part done start reading the user's input and search for it in the title.
5. Work on revealing the correct letters and displaying them.
6. Add the logic to keep track of wrong letters so they don't lose points for guessing the same letter twice.
7. After that, you can keep track of how many wrong guesses and end the game if they lose.
8. Finally, detect when they have guessed all the letters and let them know they've won!

You can also start by hard coding a single movie title in the code instead of randomly picking one from the file, then add the file reading functionality at the end.

Test often

Every time you add new code that does something new, test it.

The best way to do so is to use `System.out.println()` everytime you add new functionality to test the output of that part.

Make sure when testing to try all possible cases that you can think of (what if the user tries to guess a space character? what if they type in a number? etc)

If you test often while building your code you will end up with fewer bugs as you get closer to finishing it.

String methods

Check out all the powerful methods that Java has already written for you here.

Knowing the capabilities of your programming language can save you hours and even days of re-writing code that already exists

For example:

To find if a letter exists in a String, instead of creating a loop to compare each character you can use the `indexOf()` method which returns the position of such character in the String.