# Immutability

# Lecture Task

## - Genetic Algorithm: Lecture Task 5 -

**Functionality**: In the Song **object** implement the following method:

- A method named “costFunction” that:
  - Takes a Map of Strings to Ints as a parameter representing the rating of a specific user. The keys of this map will be youtubeIds and the values are the ratings from this user
  - Returns a function that takes a Song as a parameter and returns a Double. This function will compute the cost of a Song for this specific user based on their ratings. The function will return:
    - 1000.0 if the song has been rated with a 1 or 2 by this user. This is a very high cost relative to songs rated 3+ and will almost certainly prevent these songs from appearing in any playlist for this user
    - If the song has not been rated by this user, treat it as though the user gave it a rating of 3
    - If the song has been rated 3, 4, 5, or has not been rated and is being treated as a 3, return a cost of 1 / (bayesianRating * userRating)
      - For bayesianRating add 2 extra rating with value 3

**Testing**: In the tests package, complete the test suite named LectureTask5 that tests this functionality.

# Example

**Restriction**: No state is allowed in this question. Specifically, the keyword "var" is banned

**Question**: In a package named "functions" write an object named Numbers with a method named histogram with the following features:

- Takes a List of Ints and returns a Map of Int to Int mapping each unique Int to the number of times that Int appears in the input List

**Testing**: In a package named "tests" create a class named "TestHistogram" as a test suite that tests all the functionality listed above

# Example

**Restriction**: No state is allowed in this question. Specifically, the keyword "var" is banned

**Question**: In a package named "functions" write an object named NumberGuesser with a method named guessTheNumber with the following features:

- Plays a number guessing game where there is a hidden number that needs to be found
- Takes 2 Doubles and function Double to Boolean as parameters representing:
    - The lower bound of the number to guess
    - The upper bound of the number to guess
    - A function that tells you if you need to search higher of lower
        - This function returns true if the hidden number is higher than the input number, false otherwise
- Returns The hidden number to 2 decimal places

**Testing**: In a package named "tests" create a class named "TestNumberGueser" as a test suite that tests all the functionality listed above

# Lecture Task

## - Genetic Algorithm: Lecture Task 5 -

**Functionality**: In the Song **object** implement the following method:

- A method named “costFunction” that:
  - Takes a Map of Strings to Ints as a parameter representing the rating of a specific user. The keys of this map will be youtubeIds and the values are the ratings from this user
  - Returns a function that takes a Song as a parameter and returns a Double. This function will compute the cost of a Song for this specific user based on their ratings. The function will return:
    - 1000.0 if the song has been rated with a 1 or 2 by this user. This is a very high cost relative to songs rated 3+ and will almost certainly prevent these songs from appearing in any playlist for this user
    - If the song has not been rated by this user, treat it as though the user gave it a rating of 3
    - If the song has been rated 3, 4, 5, or has not been rated and is being treated as a 3, return a cost of 1 / (bayesianRating * userRating)
      - For bayesianRating add 2 extra rating with value 3

**Testing**: In the tests package, complete the test suite named LectureTask5 that tests this functionality.