

AITP NCC 2016 Java Competition

The Problem

Chris has decided to follow his dream of owning a business, The Entertainment Genius. He has just graduated from college and eager to put into practice everything his professors have taught him. After getting funding, Chris has determined that his top priority is to build a top of the line IT product for movie and game rentals. With little experience in technology, he decides to run a competition to see who can create the best product. He has some basic requirements that they expect the system to perform, but ultimately would like to see what special features your team can bring to the company. The project (2016AITP) contains a group of JUnit Test Cases that allow Chris to know that the basic requirements have been met. JUnit tests are the highest authority related to requirements and can be found by following the path *src/test/java*. The competition is going to provide two data files, one for movies and one for games. See some of the categories below:

- Type (movie, game)
- Title
- Rated
 - Movie (R, PG-13, G, etc.)
 - Games ((E)veryone, (T)een, (M)ature, N/A)
- User Rating (scale of 0 – 10)
- Genre (Action, Drama, Crime, Mystery, Comedy, etc.)
- Movie Format (DVD, Blu-ray, Streaming)
- Game Platform (PC, PS4, Xbox One, Wii U, etc.)
- Actors

Requirements

Core requirements:

- Allow customers to search by movie and/or game title (allow search on partial titles).
- Movie and/or game title with specific User Rating or higher sorted by release date (allow search on partial titles).
- Search movies and/or games with specific genre and rating (R, PG-13, etc.).
- Application must know the availability of the entertainment and throw appropriate exception when not available. Checkout functionality decrements stock by amount ordered only for the duration of the test (Do not store/write back to json file)

***** **Do not change anything in the JUnit tests!** *****

Bonus “Nice to have” features

Suggestions shown below in no particular order or come up with an original idea

- Bonus Credit – Do not complete any bonus features unless you have completed all the required functionality and all JUnits pass.
 - Create a Graphical User Interface (GUI), command line interface, or web UI
 - Monetary capabilities – Game or Movie rental is \$2 each
 - Discount – Rent 2 Games or Movies and get \$1 off purchase
 - Add reservation / hold ability for movie and game titles.
 - Additional search functions that will keep the customer informed

AITP NCC 2016 Java Competition

First Actions:

- Import the problem statement into your IDE.
- We have provided Maven dependency for JUnit 4. If you are not set up with the recommended IDE, you may need to add JUnit 4.
- Run your JUnit tests, code, and repeat.
 - For questions on how to run JUnits, please see document in Workspace: "How to run JUnits"

When you are done:

- Update the feedback.txt file and include the following information in it:
 - Your team ID.
 - How many JUnits you were able to execute successfully.
 - Document and describe the additional "nice to have" features included, to help the judges properly grade your submission and explain how to properly execute new enhancements.
- Export your IDE project (The judges will need your entire project to run the JUnits) as a zip file and upload on the coding competition server. If the network is unavailable, please let the proctor know and we will be able to save your project using another method.
- **DO NOT INCLUDE ANY IDENTIFIABLE INFORMATION BEYOND YOUR TEAM ID. DO NOT INCLUDE TEAM MEMBER NAMES OR UNIVERSITY NAMES.**

Rules

- The use of references such as Java APIs, are allowed for the competition.
- Contestants cannot seek help from individuals outside their team.
- Teams are expected to have the necessary Tools and Jars preloaded on their machines **prior** to the competition.
- If you believe this document and the JUnit tests conflict, the JUnit tests are the highest authority.

What the Judges will use to Grade your Submission

If your team uses something other than what is listed below, the judges will try to execute your submission but we cannot guarantee we'll be able to do so.

- Spring Tool Suite – most current version
- JDK 8 compliance

How you will be Graded

Components of submission will be weighted as follows.

- 100% core requirements met, including:
 - Number of JUnits that pass using correct functionality in the program
 - Maintaining Object Oriented Programming principles
 - Code documentation
 - Code must compile and execute
- Do not complete any Bonus unless you have all the JUnit tests completed
 - Bonus credit awarded for any extra features added (up to 20%)