**Names:** Adam Kraus, Chami Senerath, Hunter Hufstutler

**Language:** Typescript

**Coding environment**: NodeJS w/ TypeScript 4.5.2

### **Distinct pieces of functionality (that the team personally wrote for the project part):**

1. Database Connection
   1. /src/data-source.ts
   2. Everything in /src/entity
2. Management
   1. /src/scripts/ts/add.ts
   2. /src/scripts/ts/list.ts
3. Searching
   1. /src/scripts/ts/watchlist.ts

### **Section 1: Comparison to C++, Java, Python, and/or Scala**

1. **What is the language’s philosophy?**

JavaScript with type safety

1. **Compare and contrast your language in terms of the location it is used.**

<answer here>

1. **Compare and contrast your language in terms of where it excels and where it fails**
   * **Excels**

<answer here>

* + **Fails**

<answer here>

1. **Compare and contrast your language in terms of portability, simplicity, orthogonality, AND reliability.**

<Note: maximum of one page, and subheadings are suggested>

* + **Portability**

<answer here>

* + **Simplicity**

<answer here>

* + **Orthogonality**

<answer here>

* + **Reliability**

<answer here>

### **Section 2: Syntax, OOP**

1. **A) Write an example of one type of assignment expression in the language.   
   B) Then write the generic format of an expression. Eg. Pre-fix, post-fix, curly brackets, indentation requirements, etc. The goal here is the general appearance of a line or block of code.**

<answer here.>

1. **A) How does the language support extension, etc. (single inheritance, interfaces, root object, class OOP, prototype OOP, other OOP, file importing, file extension, plugins, piping, module linking, etc.)?   
   <**answer here>

**B) Give an example.**

<answer here. Note: Unless it is esoteric, there will be some extensibility. If you say it does not support class-based OOP (the variety used almost exclusively in this class), so there is no extensibility, I will not believe you as that means everything must be done in one file! ***Bash*** supports extensions! Max at 2 options.>

1. **A) How does the language handle module/namespace/packages/etc.?**

<answer here>

**B) What is the scope operator(s)? Alternatively, explain how to pick which variable if two code courses contain the same name?**

<answer here. Note: the scope operator in C++ is ::. In Java, it is . Python has a few options. >

**A) Does the language allow function overloading (name repetition), function redefinition, and/or function overriding?**<answer here>

**B) Give example syntax if it does.**

<answer here. Note: redefinition and overriding are NOT the same thing. >

### **Section 3: Binding, type system, and data type range**

1. **Is the language static or dynamically typed? Give example syntax in code.**

Static typing

1. **Is the language static or dynamically scoped? Give an example in code.**

<answer here>

1. **Describe the type system (equivalence, compatibility, etc.).**

<answer here>

1. **What are the built-in data types and their ranges? (list 4-10, or send me a note if you believe that there are less than 4)**

<answer here>

### **Section 4: Control flow, functions, specialties**

1. **What are the selection and repetition structures of the language, and what are their syntax?**

<answer here. >

1. **Are functions pass-by-value, pass-by reference, etc.? Give example syntax in code.**

<answer here>

1. **Describe at least two of the language specialties.**

Specialty one:

<answer here>  
Specialty two:  
 <answer here>