OBJECT ORIENTED PROGRAMMING

JavaScript as development platform

Developed almost 20 years ago, JavaScript has outstanding expressiveness and flexibility allowing developers to choose how and what to program with JavaScript rather than dictating a way it meant to be used by its creators. What makes JavaScript unique is prototypal inheritance and lack of traditional notion of classes (fixed schema for objects). Dynamic class-free type system of JavaScript allows to not only use prototypal inheritance, but also practices brought from other languages. The goal of this assignment is to familiarize yourself with implementation of Object Oriented Programming in JavaScript and develop practical skills in design and implementation of object hierarchies.

Problem

Design and implement Client-Side Logging Library in adherence to following user stories:

1. As a developer I want to have an ability to log information.
2. As a developer I want to have an ability to send log information to the console.
3. As a developer I want to have an ability to send log information to an alert window.
4. As a developer I want to have an ability to send log information to the current window.
5. As a developer I want to have an ability to send log information to abstract Web API endpoint.
6. As a developer I want to have an ability to select from built-in logging methods.
7. As a developer I want to be able to extend the library with custom logging methods.
8. As a developer I want to library to automatically log all unhandled client-side exceptions.
9. As a developer I want to have an ability to extend library with my own instrumentation methods so that the library can monitor not only exceptions but other events too.
10. As a developer extending the library with my own logging or instrumentation methods I want to have established contracts in a form of base class of API documentation so that I can easily know how to extend the library.
11. As a developer extending the library I want to see clear errors and warning in case if I violate the contract of logging or instrumentation methods.
12. As a developer I want an ability to access the library through global variable, AMD module or CommonJS module.

Significant implementation details & design goals:

1. Architecture of the library is what matters most. Following SOLID principles is crucial to successfully manage dependencies in your code.
2. Extensibility matters.
3. Use your existing knowledge to design clean API for your library. But keep in mind that API design is always a tradeoff: API that good for everybody, good for no one.
4. Don’t take word ”method” literally when reading user stories. In your implementation ”method” can be mapped to construct of JavaScript language or OOP concept.