Assignment 2: Secure Web Programming and Node.js, Question 2 (vending machine)

> Consider the following JavaScript code for the vending machine we did in class. The design and implementation are rife with bad practices. Please do the following:
>
> 1. Identify all issues and explain why they can present security concerns. Explain the fix. Please upload your document as a PDF file.
> 2. Fix all of the issues and submit the corrected .js file.

Introduction to JavaScript and Secure Coding

Issue 1: All numbers are internally represented using double-precision 64-bit format (IEEE 754)

Security Concern: mathematical calculation errors

Fix: Use big-js library (and/or convert decimals to integers prior to performing mathematical calculations)

Open Terminal

```
cd vending_machine/
```

Install big.js

```
npm install big.js
```

Use big-js on any floating point variables.

Issue 2: Passing Invalid arguments to parseInt or parseFloat

Security Concern: Will lead to incorrect math calculations, allows program input to be exploited.

Fix: Validate User input

Use `readlineSync.questionFloat()` for float inputs. Readline-sync has more available utility methods.

Issue 3: Using var instead of `let` or `const` to declare variables

Security Concern: Will lead to confusing scope issues because var declarations are automatically hoisted to the top of the program.

Fix: use Let or const instead, avoid using var.

Issue 4: `==` performs type conversions and then compares

Security Concern: Returning incorrect boolean due to type conversion

Fix: Generally you should always use `===` or `!==`

Issue 5: Not using prototype (or syntactic sugar provided by Classes) functions for vending machine object

Security Concern: Vulnerable to DDOS attack due to heavy memory usage.

Fix: Use prototype for the vending machine object methods. Each vending machine object will share the same methods in memory instead of creating duplicate methods for each new vending machine instance. Another possible solution would be to use the syntactic sugar provided by ES6 Classes to create objects from prototypes.

More Secure Coding

Issue 6: JavaSript tries to be helpful and fix or ignore programming errors.

Security Concern: This can lead to security vulnerabilities and unintended logic errors

Fix: use Use Strict Mode to use a stricter version of JavaScript. Strict Mode changes some silent errors to throw exceptions, among other things.