**Craft Demonstration**

**Vending Machine Challenge**

**Overview:**

Intuit’s Tax Platform Development Team would like to invite you to take our Vending Machine Craft Demonstration Challenge. We have provided some information at the bottom on how to return the data. Any questions on the Craft Demo, please reach out to Shane and for all other questions, please reach out to your recruiter.

**Problem:**

* Design and code an implementation of a vending machine that can handle different products, can accept payments, and can vend product.
* You are designing software for a variety of real vending machines that handle different product types. Keep this in mind as you design. For example, your design for dispensing the product would need to be able to handle the logic that drives the mechanical aspects of various types of machines. For the demonstration, your dispense mechanism would dispense to the system console and/or a mock class. In future production implementations, the dispense mechanism may drive different types of motors that dispense the product.
* For this example, your user interface for the vending machine should be the system console.
* Depending on your assumptions there may be other external communication points (like a mechanical motor used in dispensing products) that are not feasible to code as part of this exercise. If so, mock those up accordingly.

**Notes:**

* Formal documentation of the design is not required. However, feel free to document anything you feel you can better explain through diagrams and more formal documentation.
* Document any assumptions. Example: If you assume the vending machine should be able to handle multiple payment types such as coin or credit cards, document this.
* Your project should be as close to production quality as possible in the allotted time frame for this exercise.
* Intuit has a predominantly Java backend, as such, the people who will be reading your code “speak” Java better than any other language. If you solve the problem in a language other than Java; any language specific idiosyncrasies should be well documented in either the comments or the README.

**Please deliver the code to us this way:**

**Option 1: GitHub**

* Check your code into a public repo
* Create a GitHub release when you submit it to us (we specifically want to see what you had accomplished in the elapsed time).

**Option 2: Email to Recruiter**

* Zip up your project

**Questions pertaining to our Craft Demo:**

* Contact: Shane Crouch
* eMail: [shane\_crouch@intuit.com](mailto:Vishwa_Krishnamurthy@intuit.com)
* Phone: (512) 800 0065

**Please send your Craft Demo information to your recruiter when done.**

*Thank you for taking time to work on our Craft Demonstration Challenge!*