

# ***CSC 413 Project Documentation***

***Fall 2019***

***Wameedh Mohammed Ali***

***920678405***

***CSC 0413.02***

***GitHub Repository Link***

***[https://github.com/csc413-02-](https://github.com/csc413-02-FA2020/csc413-p2-Wameedh/tree/master/interpreter)***

***[FA2020/csc413-p2-](https://github.com/csc413-02-FA2020/csc413-p2-Wameedh/tree/master/interpreter)***

***[Wameedh/tree/master/interpreter](https://github.com/csc413-02-FA2020/csc413-p2-Wameedh/tree/master/interpreter)***

## Table of Contents

1	Introduction .....	3
1.1	Project Overview .....	3
1.2	Technical Overview .....	3
1.3	Summary of Work Completed .....	4
2	Development Environment .....	4
3	How to Build/Import your Project .....	5
4	How to Run your Project .....	5
5	Assumption Made.....	6
6	Implementation Discussion .....	7
6.1	Class Diagram.....	7
7	Project Reflection .....	7
8	Project Conclusion/Results .....	7

# 1 Introduction

## 1.1 Project Overview

This project is an interpreter of a computer programming language X. The purpose of it is to read files of programs written in X then interpret the code to byte code that on a virtual machine. In other words, it would allow us to write computer code in language X that is more user friendly then the it would take that and changed it a byte code which is machine friendly.

## 1.2 Technical Overview

The program runs on a virtual machine which means it could work on any environment. The project is written in Java.

The project contains one main modules, that is:

- Interpreter

The interpreter module contains two modules and three files, these are:

- Bytecode: This Module contains the following files:
  - ByteCode.java
  - BranchCode.java
  - HaltCode.java
  - PopCode.java
  - FalseBranchCode.java
  - GotoCode.java
  - StoreCode.java
  - LoadCode.java
  - LitCode.java
  - ArgsCode.java
  - CallCode.java
  - ReturnCode.java
  - BopCode.java
  - ReadCode.java
  - WriteCode.java
  - LabelCode.java
  - DumpCode.java
- virtualMachine: This Module contains the following files:
  - Program.java
  - RunTimeStack.java
  - virtualMachine.java
- ByteCodeLoader.java
- CodeTable.java
- Interpreter.java

### 1.3 Summary of Work Completed

I have implemented all the Supported ByteCodes including the abstract ByteCode and BranchCode classes. I have implanted all the subclasses of ByteCode and BranchCode these are:

1. Halt ByteCode
2. Pop ByteCode
3. FalseBranch ByteCode
4. Goto ByteCode
5. Store ByteCode
6. Load ByteCode
7. Lit ByteCode
8. Args ByteCode
9. Call ByteCode
10. Return ByteCode
11. Bop ByteCode
12. Read ByteCode
13. Write ByteCode
14. Label ByteCode
15. Dump ByteCode

I have modified the fallowing files:

- ByteCodeLoader.java
- VirtualMachine.java
- RunTimeStack.java
- Program.java

## 2 Development Environment

The development environment I used is IntelliJ IDEA on a MacBook Air laptop with macOD Catalina with Java version 13

### 3 How to Build/Import your Project

To import the project clone it from GitHub then make sure you follow the instructions provided in setup section in the documentation provided.

“When importing The interpreter project you will use the root of your repository as the source.

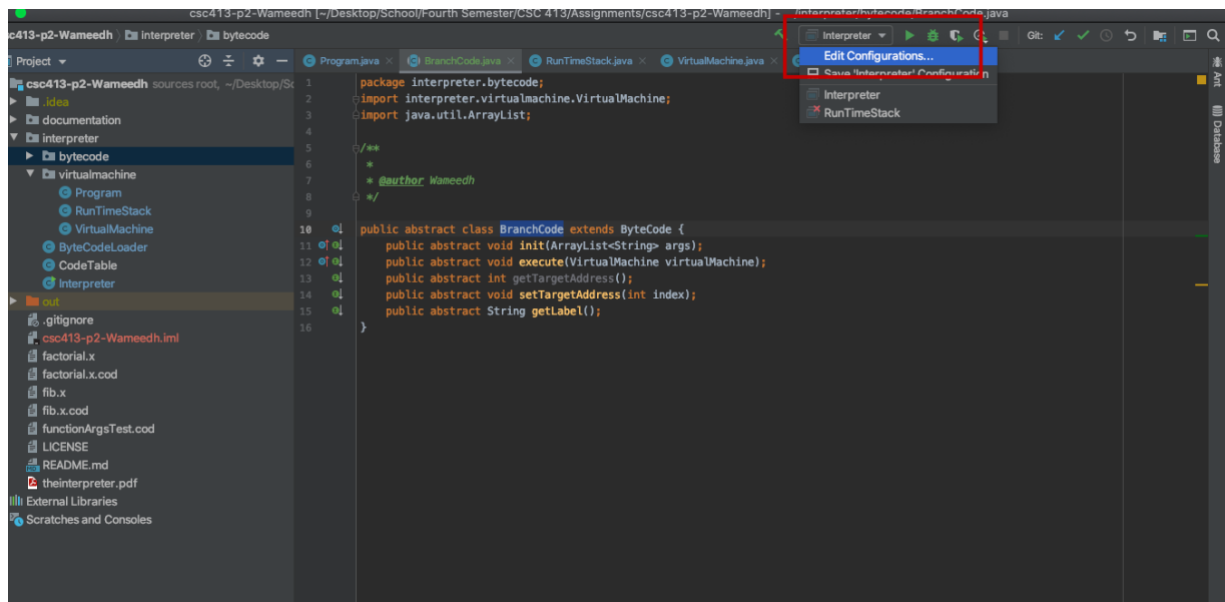
DO NOT USE the interpreter folder as the root. This will cause the project to not work

and in the end force you to change the file structure.”

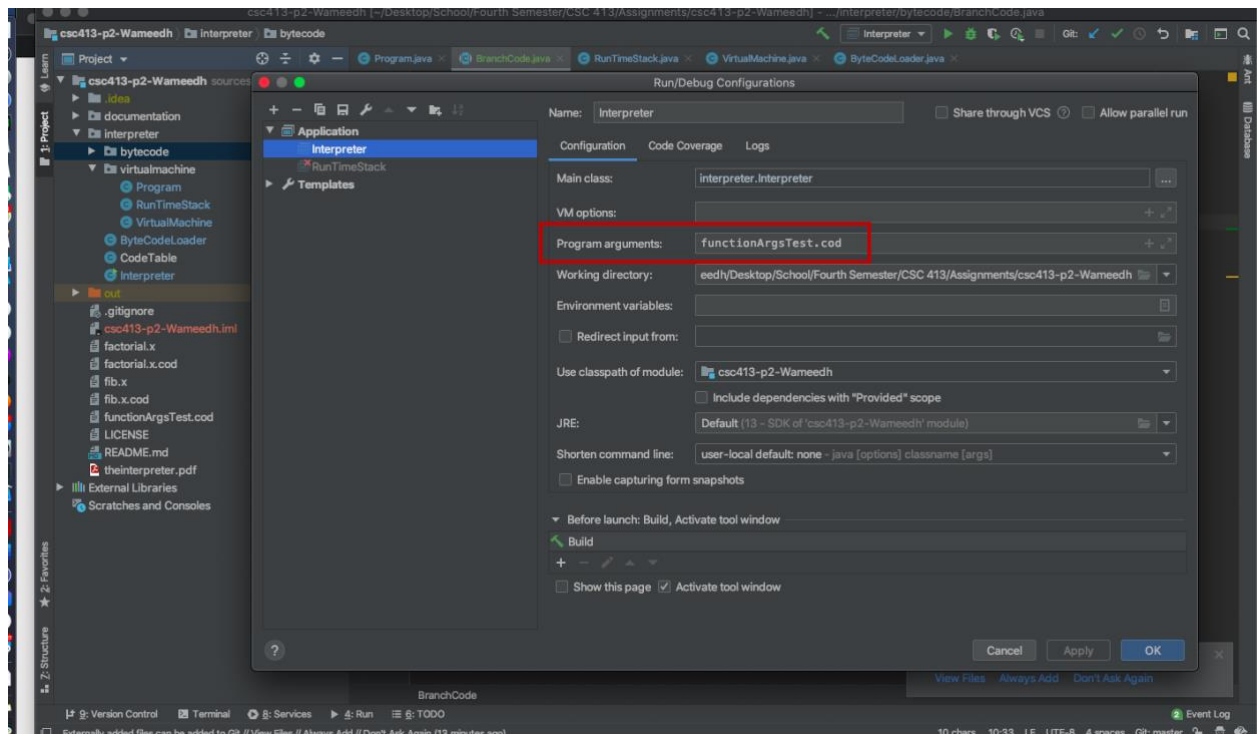
After importing the project successfully, you can build the project as instructed in the next section.

### 4 How to Run your Project

This project was built and tested using IntelliJ IDEA. To Run Select “Edited Configurations” as shown below:



Then add the name of the file to be tested in the program arguments section as shown below:



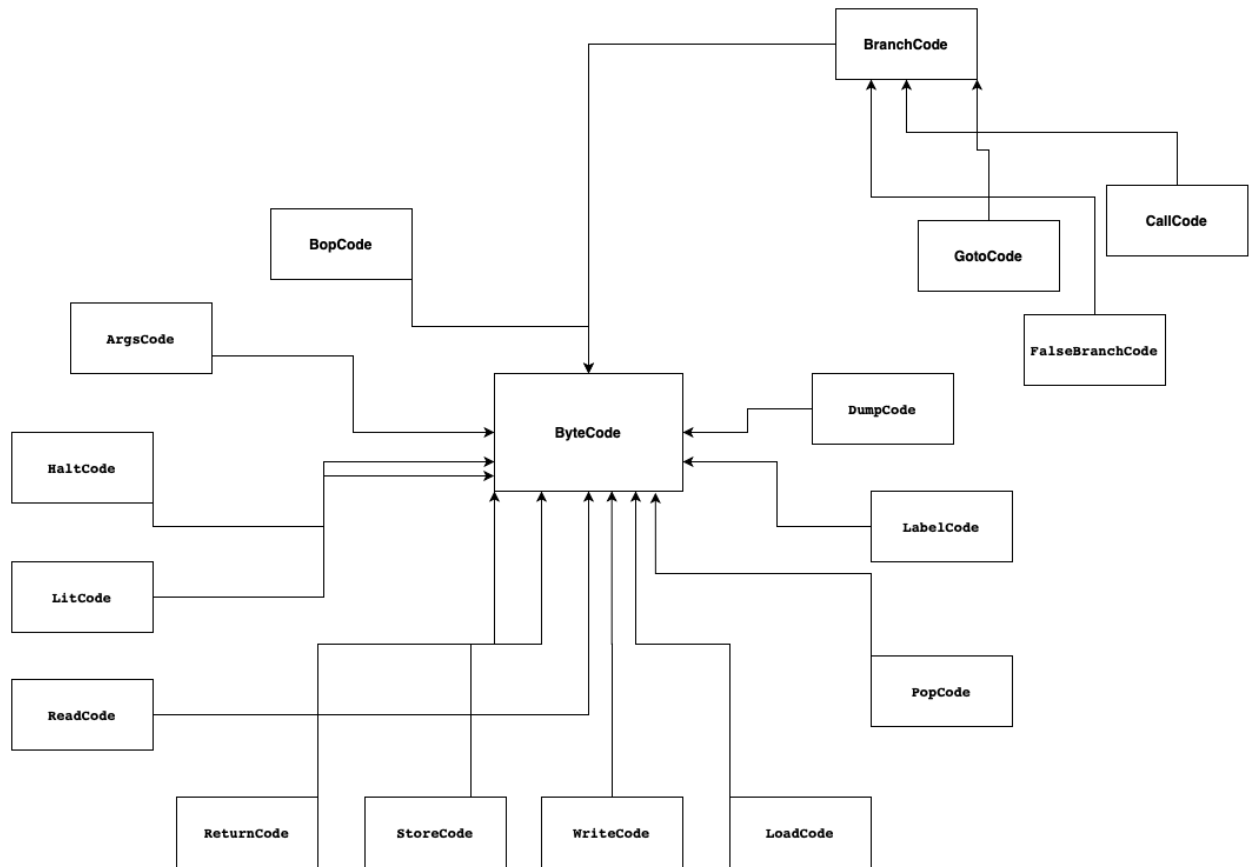
Then run the project from the “run” button.

## 5 Assumption Made.

I assumed that all values passed are integers and that the implementations in CodeTable.java and interpreter.java are correct and need no change.

## 6 Implementation Discussion

### 6.1 Class Diagram



## 7 Project Reflection

The project was very hard to understand, up until now I can be 100% sure what I am implementing. I don't feel confident enough to explain the project. I started the project earlier than the previous project to avoid the experience that I had when I ran out of time. Unfortunately, the project due date fall in a time where I had so much schoolwork and life changing events. I did my best, but I regret that I couldn't get the project to run.

## 8 Project Conclusion/Results

The project unfortunately run but throw an error:

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 1 out of bounds for length 0
```

I think if I can get some more time I would be able to get to work and does the job. Unfortunately, I don't have more time.