

***CSC 413 Project Documentation***  
***Summer 2020***

***Amit Joshi***

***918210532***

***CSC 413.01***

<https://github.com/csc413-01-SU2020/csc413-p2-amitkhalsa25>

## Table of Contents

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

# 1. Introduction

## 1.1. Project Overview

In this project, we have new Language called Language X. Made also interpreter to read and run the files in the new language being used. Extends the language to x.cod as well and I made Virtual Machine that works with the interpreter to run the files. The interpreter comes with code that has execution for each function in the language so code works then and can be written in language X!

## 1.2. Technical Overview

In this project I implemented an interpreter for mock language X which can be the simplified version of java as well. For the interpreter, it should go and proceed the byte codes that are created from the source code with extension x. Then for the interpreter and other java files like virtual machine collaborate together to run the program written In the language X. Files have extension x.cod including fib.x.cod.

## 1.3. Summary of Work Completed

I have implemented RunTimeStack class that records and process the stack of frames. Virtual Machine class is also used to operate the program. Also it contains abstract functions for the byte code. I implemented the program class as well which stores the byte code from the source file and stores code in Array List as well all through. I implemented the resolve address label function that resolves the symbolics used in the functions. ByteCodeLoader class was also Implemented and it loads the byte codes from the file into an array list. I created other byte code classes as well including halt, pop, goto, store, write, return code, load, call, label, false branch, Bop, and lit, dump and helps excute the program.

# 2. Development Environment

Version Used: Java 11

Ide: IntellaJ

# 3. How to Build/Import your Project

Copy the link, go to the terminal, then git clone paste the link. Then open intellaJ and click open or import option and repo is cloned choose your folder and then click ok option.

# 4. How to Run your Project

To run the project you go to the Interpreter class and edit the configuration and change the program arguments to fib.x.cod or factorial.x.cod.

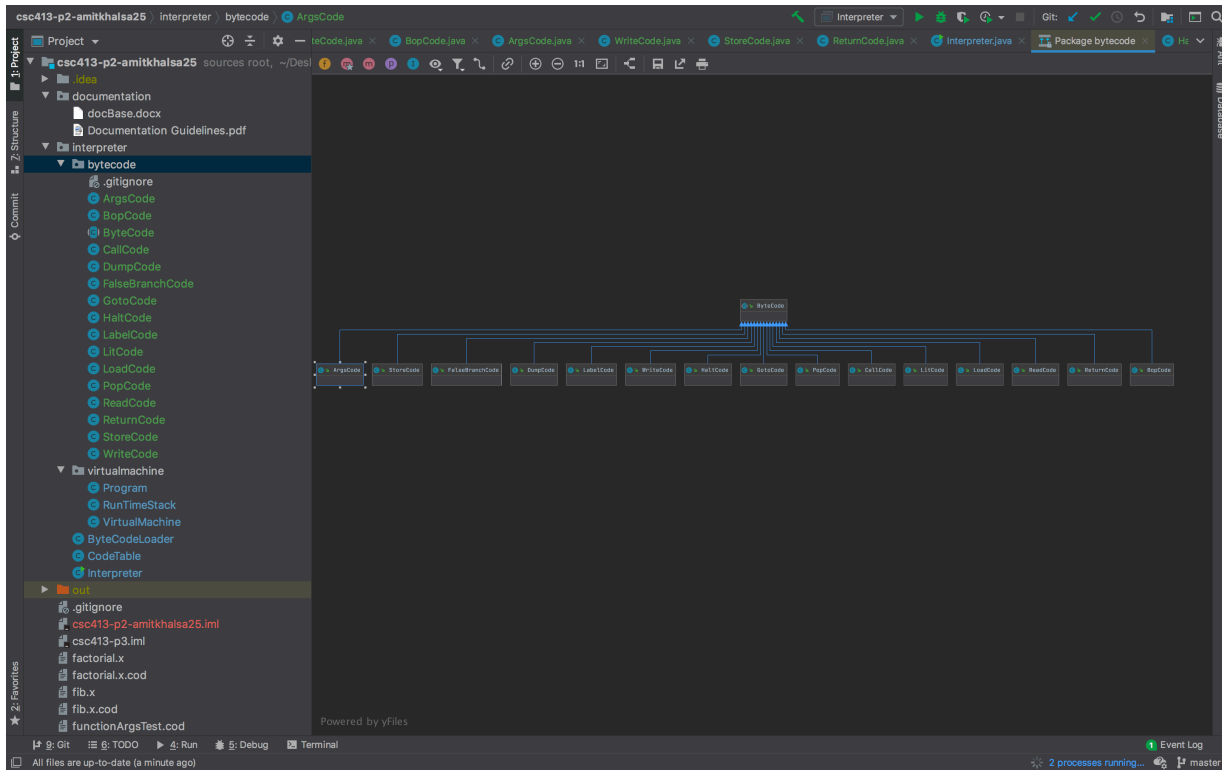
# 5. Assumption Made

I assume ByteCode source program like .cod files are to test.

# 6. Implementation Discussion

When I implement the project, I looked at first file we did and in our TableCode is where we first start off making subclasses that we needed to use.

## 6.1. Class Diagram



## 7. Project Reflection

Overall, this is one of my hardest projects I have ever done. While I watched the ASMT 2 videos it helped a lot more like assignment one where we need to build our subclasses. I was having hard time getting fib.x.cod to run but I worked it out and everything else was perfect so I hope I get good grade. I also deleted AddressLabel subclass but I really didn't need it anymore.

## 8. Project Conclusion/Results

Project was fun very interesting and helpful for my coding skills to build up. Every project everyone can improve! I put lot's effort in this project as well. Everything works but just little error towards the end but I figured it out and finished but other then that it was fun!