**Programming Assignment 1 Part A**

***Solution Description:***

My program simply does a module pass and iterates over the functions, basic blocks, and instructions of the program and outputs their information. The output will be a list of functions, where each function will list its basic blocks, and each basic block will list its instruction. For each function, it checks whether it is a prototype or definition and prints out the corresponding “declare” or “define” before then printing the function signature (the return type, name, and arguments). If the function has basic blocks, then my program will iterate over each basic block, printing the block’s label and then subsequently iterate over the block’s instructions and print each instruction’s source code line number. This is repeated for each function until all functions are outputted. My program makes use of the Module, Function, BasicBlock classes and their iterators and functions to get the information necessary to identify and print the needed information.

***Structure of Archive:***

The structure of the archive resembles closely to the skeleton project provided. The pass is given as a list of functions and their information. Each function is separated by an empty line and all information of the function, if defined, is contained within the curly brackets “**{***…body…***}**.” In the body contains every basic block in the function and each block begins with a label number and ends with a new line. Each basic block will print the corresponding LLVM instructions along with the source code line numbers before each instruction. Below is the format for function prototype and definition.

Function Prototype:

**define** *<return type>* **@***<function name>***(***<parameter type>, <parameter type>, …, <parameter type>***);**

Function Definition:

**define** *<return type>* **@***<function name>***(***<parameter type>, <parameter type>, …, <parameter type>***){**

**; <label>: %**<*label number*>

<*Source Code Line Number*> *<LLVM instruction>*

*… …*

*<Source Code Line Number>* *<LLVM instruction>*

**; <label>: %**<*label number*>

<*Source Code Line Number*> *<LLVM instruction>*

*… …*

*<Source Code Line Number>* *<LLVM instruction>*

**}**

**Bold**: independent of function (text always present for all functions)

*<Italics>*: dependent on function (text values depend on function)