The George Washington Universityshort line



**Computer System Architecture (CSCI 6461\_10)**

Project Part 0 User Guide

**Submitted By:**

Group 9

Abhiyan Sainju

Boxin Yang

Udaykiran Kalyanapu

**Submitted to:**

Professor Morris Lancaster

# Project Part 0 Assembler: User Guide

## Prerequisites

* Java17 or later version installed.
* Java Runtime Environment (JRE) or Java Development Kit (JDK) installed.
* Text file with assembly instructions prepared.

## Usage

* Open your command line interface.
* Navigate to the directory where the Assembler program is located.
* Execute the program by simply clicking on the run.jar file or
* Type in **java -jar run.jar ‘your file name’**

p.s. If the file name is not provided, by default it runs test1.txt.

This will start the Assembler program.

1. **Input File:**

By default, the Assembler expects a file name from command line. If the file name is not provided, it will read from “test1.txt”.

1. Output File:

The Assembler will generate two files:

* *ListingFile.txt*: This file contains a list of the processed instructions along with their original lines.
* *LoadFile.txt*: This file contains the processed instructions ready for loading into memory.

## Detailed Function Descriptions

* *\*\*loadDict()\*\*:* Initializes the opcode to binary and registers to binary mappings.
* *\*\*readFile()\*\*:* Reads assembly instructions from the input file.
* *\*\*BinaryToOctal()\*\*:* Converts binary strings to their octal representation.
* *\*\*formatAndConvertInstruction()\*\*:* Formats the instruction components and converts them to octal.
* *\*\*handleMiscellaneousInstruction()\*\** and other `handle` methods: Process different types of assembly instructions.
* *\*\*parse()\*\*:* Parses the instructions, categorizes them, and converts them to the output format.
* *\*\*writeListingFile()\*\*:* Writes the detailed listing of instructions to a file.
* *\*\*writeLoadFile()\*\*:* Writes the load-ready instructions to a file.
* *\*\*cleanup()\*\*:* Cleans up the load file by removing blank lines.
* *\*\*run()\*\*:* Orchestrates the reading, parsing, and writing of instructions.
* *\*\*main()\*\*:* The entry point to run the Assembler program.

**Troubleshooting:**

* *\*\*Missing Input File\*\*:* If the program cannot find the input file, check the filename and path.
* *\*\*Output Files Not Generated\*\*:* Make sure you have write permissions in the directory.
* *\*\*Java Exception\*\*:* If you encounter a Java exception, check that your Java version matches the version used to compile the Assembler class.