1. **Main data structures :**
2. **OBTable :**

* Uses unordered\_map to hold Opcode for all operations.

**methods :**

isOp: returns true if string provided is an assembler operation.

getOpCode :returns the integer OpCode corresponding to the provided string.

1. **SymbolTable :**

* Uses unordered\_map to hold Names of symbols as a key and a list , that holds where they were used ,as a value until their address ,at where they are kept in memory, is defined.
* **methods :**
* request : adds a new request for the undefined symbol to the list of its addresses

,throws a runtime error if the symbol was already defined and a request can’t be added.

* Define : defines address for where a symbol is kept in memory,throws a runtime error if symbol was already defined in memory.
* Contains : returns true if symbol was defined in memory.
* Get : return the address of a symbol kept in memory,throw runtime error if symbol wasn’t defined in memory yet.
* Finish : called when end is found in assembled program,checks if all symbols in map was defined properly in memory,throws a runtime error if a symbol wasn’t defined yet.

1. **Registers :**

* Uses unordered\_map to hold Opcode for all registers.
* Methods **:**
* isRegister: return true if provided string is a register.
* getRegister : return integer opCode corresponding to the provided register name.