

Task 4.1 - Basic Machine Code Translator (12 points)

Write code to implement the basic parsing and translation of A and C instructions.

- Use the provided template files in your preferred programming language
- Complete the following methods:
 - `doSecondPass/generateMachineCode`
 - `parseInstructionType`
 - `parseInstructionDest`
 - `parseInstructionJump`
 - `parseInstructionComp`
 - `parseSymbol`
 - `translateDest`
 - `translateJump`
 - `translateComp`
 - `translateSymbol`
- You may add methods, but do not modify the provided method signatures
- You do not need to implement the Symbol Table or L-instructions
- Submit your completed source and test files in the same directory as your files from Parts 1-3.
- Only submit files for 1 programming language.

Test Cases:

- Write at least 3 test programs.
- A sample test program is provided in `Translator100.asm`
- Each test case should be in a file named `Translator1XX.asm` where `XX` is a number starting at `01`.
- Your mark for this task may be **scaled down** for poor/missing testing.

Task 4.2 - Full 2-Pass Assembler (12 points)

Write code to implement the first pass and Symbol Table.

- Use the provided template files in your preferred programming language
- Complete the `SymbolTable` class and `doFirstPass` methods, as well as updating the `translateSymbol` method.
- You may add methods, but do not modify the provided method signatures
- Submit your completed source and test files in the same directory as your files from Parts 1-3.
- Only submit files for 1 programming language.

Test Cases:

- Write at least 3 test programs.
- A sample test program is provided in `Translator200.asm`
- Each test case should be in a file named `Translator2XX.asm` where `XX` is a number starting at `01`.
- Your mark for this task may be **scaled down** for poor/missing testing.