

## Intro to Computer Systems :: Project 8: VM Translator II

Student name: \_\_\_\_\_

### **Grading method**

In evaluating the second part of your VM Implementation program (AKA Translator) we look into the following things:

- **Code quality:** since the software design was provided, your code will be judged mainly for elegance and readability.
- **Correctness:** your translator must translate correctly the supplied VM test programs: FibonacciElement and StaticTest. "Correct translation" occurs when the Translator generates Hack code that, when run on the CPU Emulator, generates the desired results.
- 

| <i>Assembler</i> |       | <i>Comments</i>   |
|------------------|-------|---|
| Packaging        | / 5   | Directory (folder) with your name on it, containing (1) /src directory of your code, (2) a README file with instructions for compiling and running your program, and comments on things that don't work and how you tried to fix them.  |
| Working?         | / 45  | Does the program generate assembly output that will work on all test inputs?  |
| Well built?      | / 50  | 14 points for documentation. But, don't over-document! For every method you write, document what it does, what parameters it takes, and what it returns. Use your judgment to add more documentation when needed.<br><br>16 points for a VM translator that produces efficient assembly code. Generally speaking, the fewer the number of generated assembly instructions, the better.<br><br>20 points for a good and clean implementation that we can easily read and understand. |
| Total            | / 100 |   |

Total grade: \_\_\_\_\_