1. What are the two major questions this course strives to answer? (2)

2. What is the difference between a fixed-program device and a stored-program device? (2)

3. What is abstraction, and what are its benefits? (2)

4. State true or false:
   a. If I plan to use a different programming language, I would have to develop a new ISA. (2)

   b. If I disallow some instructions from my ISA, some Python instructions may stop working.

5. What is an algorithm? What are the two properties an algorithm must satisfy? (2)

6. Design a list of steps similar to the square root algorithm that computes A*B and stores the result in C, where A and B are positive integers. Assume that the only steps allowed are adding, comparing (> , <, >=, <=, =), and storing. You are allowed to create a temporary variable, overwrite A, or overwrite B. (2)
Now verify that your algorithm works by testing the values $A=2$ and $B=3$. Iterate through the steps until completion. Keep track of all $(A, B, C,$ and any temporary variable) values by writing them down at the end of each step. (2)

7. Explain algorithmic complexity and how trade-offs are involved in choosing an algorithm. (2)

8. Programming is usually done using high-level languages as that is much easier than using ISA instructions directly. But, what is a benefit in programming using ISA instructions directly? (1)

9. In some programming language, to display the natural logarithm of any number $n$, this is what we need to write: \textit{print} (\textit{log}(n)) (2)
   A group of students were assigned to display \textsf{log}(5). Three different students wrote:
   a. \textit{print} (\textsf{log}(5))
   b. \textit{print} (\textsf{log}((5))
   c. \textit{print} (\textsf{log}(55))
   d. \textit{print} \textsf{log}(5)
   Which student got the right answer? What were the type of errors experienced by the other three students?

10. What does Turing Completeness mean? (1)