Instructions:

1) This test is 5 pages in length.

2) You have 40 minutes to complete and turn in this test.

3) Short answer and essay questions include guidelines for how much to write. Respond in complete English sentences. Responses will be graded as described on the syllabus.

4) This test is closed books, notes, papers, smartphones, laptops, friends, neighbors, etc.

5) Use the backs of pages in this test packet for scratch work. If you write more than a final answer in the area next to a question, circle your final answer.
1. [15 points]
Describe an example policy P that may be enforced by a type checker, and explain how a type checker can enforce P, at the level of detail discussed in class. [2-4 sentences]

2. [4 points]
What are Type I and Type II errors? [1 sentence]

3. [15 points]
Why do we separate policies from mechanisms? What is the relationship between policies and mechanisms? [2-4 sentences]
4. [4 points]
What is dead code? [1 sentence]

5. [20 points]
Compare and contrast static and dynamic mechanisms. State the primary advantages and disadvantages of each. [1 paragraph]
6. [7 points]
Complete the diagram below by drawing the subsets of policies discussed in class.

7. [35 points]
For each of the following policies, write “pol” if it is a non-property policy, “prop” if it is a non-safety and non-liveness property, “S” if it is a safety property, and “L” if it is a liveness property. Also write 1-2 sentences for each policy, explaining the intuition behind your answer.

\[ P_1 = \{ \{ t^1, t^2, \ldots \} \mid \forall i: t^i \text{ is infinite} \} \]

\[ P_2 = \{ \{ t^1, t^2, \ldots \} \mid \forall i, j, n, m: (input(n) \in t^i \land output(m) \in t^j \land input(n) \in t^j) \Rightarrow (output(m) \in t^j) \} \]
$P_3 = \{ \{ t^1, t^2, \ldots \} \mid \forall i : \epsilon \leq t^i \}$
(where $\epsilon$ is the empty trace, which is a prefix of every trace)

$P_4 = \{ \{ t^1, t^2, \ldots \} \mid \forall i : write(5) \not\in t^i \}$

$P_5 = \{ \{ t^1, t^2, \ldots \} \mid \forall i, j, t:
\quad (t; request-resource(j) \leq t^i) \Rightarrow (t; request-resource(j); send-resource(j) \leq t^i) \}$
(Every request for resource $j$ must be followed by a send of that resource.)