• HW0 due tomorrow (Friday 10/1 by 5PM).

• Any questions before we dive in?
  – What are the most interesting/confusing/puzzling things so far in the course?
Agenda

- Introductions?
- Website tour
- Review and practice logical reasoning about code with Hoare Logic
- Review and practice logical strength of assertions (weaker vs. stronger)
- Onwards to forward reasoning
Introductions
Website Tour
Why reason about code?

• Prove that code is correct
• Understand *why* code is correct
• Diagnose *why/how* code is *not* correct
• Specify code behavior
From lecture:

**Hoare Logic: First definitions**

- **Program State:** Values of all related variables
- **Assertion:** True/False claim (proposition) about the program state at a certain point in execution
- An assertion holds for a program state if it is true at that point.
- **Precondition:** Assertion before the code
  - Assumptions about when the code is used
- **Postcondition:** Assertion after the code
  - What we want the result of the code to be
Hoare Logic

- **Hoare Triple**: Two assertions surrounding a piece of code
  - $\{\{ P \} \} S \{\{ Q \}\}$
    - $P$ is the precondition, $S$ is the code, $Q$ is the postcondition
    - $P, Q$ are *specifications*
- A Hoare triple $\{\{ P \}\} S \{\{ Q \}\}$ is **valid** if in any state that $P$ holds, $Q$ holds after running the code $S$.
  - If $P$ is true, after running $S$ we have that $Q$ is true.
  - Otherwise the triple is **invalid**.
Let’s practice!
(Q1, Q2)
A Note on Implication (=>)

- Implication might be a bit new, but the basic idea is pretty simple. Implication p => q is true as long as q is always true whenever p is

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p =&gt; q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Weaker / Stronger Assertions

- If $P_1$ implies $P_2$ (written $P_1 \Rightarrow P_2$) then we say:
  - $P_1$ is **stronger** than $P_2$
  - $P_2$ is **weaker** than $P_1$
- In other words:
  - $P_1$ is “more difficult” to satisfy than $P_2$
  - $P_1$ puts more constraints on program states
  - $P_1$ gives us more information about the program state
Let’s practice!
(Q3)
Forward Reasoning
Forward Reasoning

- “What facts follow from initial assumptions about the code?"
- Precondition is *given*
- Fill in the *strongest* postcondition
  - For an assignment statement $x = y$
    - Add fact “$x = y$” to what is known
    - important subtleties here (more later...)
  - Later: if statements and loops
From lecture:

Example of Forward Reasoning

\[ \{\{ w > 0 \}\} \]
\[ x = 17; \]
\[ \{\{ w > 0 \land x = 17 \}\} \]
\[ y = 42; \]
\[ \{\{ w > 0 \land x = 17 \land y = 42 \}\} \]
\[ z = w + x + y; \]
\[ \{\{ w > 0 \land x = 17 \land y = 42 \land z = w + 59 \}\} \]
Let’s Try One Together (Q4)

Suppose we know that $i \geq 2$ at the start…

$$\{ \{ i \geq 2 \} \}$$

\[
x = 2 \times i;
\]

\[
y = x;
\]

\[
z = (x + y) / 2;
\]
Let’s Try One Together (Q4)

Suppose we know that $i \geq 2$ at the start, what do we know about $z$ at the end?

```plaintext
{
  i >= 2
}

x = 2 * i;
{
  x = 2 * i \land i >= 2
}

y = x;

z = (x + y) / 2;
```
Let’s Try One Together (Q4)

Suppose we know that $i \geq 2$ at the start, what do we know about $z$ at the end?

\[
\begin{align*}
\{ \{ i \geq 2 \} \} \\
x &= 2 \times i; \\
\{ \{ x = 2 \times i \land i \geq 2 \} \} \\
y &= x; \\
\{ \{ y = x \land x = 2 \times i \land i \geq 2 \} \} \\
z &= (x + y) / 2;
\end{align*}
\]
Let’s Try One Together (Q4)

Suppose we know that $i \geq 2$ at the start, what do we know about $z$ at the end?

$$
\{ \{ i \geq 2 \} \} \\
x = 2 * i; \\
\{ \{ x = 2 * i \land i \geq 2 \} \} \\
y = x; \\
\{ \{ y = x \land x = 2 * i \land i \geq 2 \} \} \\
z = (x + y) / 2; \\
\{ \{ z = (x + y) / 2 \land y = x \land x = 2 * i \land i \geq 2 \} \}
$$
Let’s Try One Together (Q4)

Suppose we know that $i \geq 2$ at the start, what do we know about $z$ at the end?

$$\{\{ i \geq 2 \}\}$$

$x = 2 \times i$;
$$\{\{ x = 2 \times i \land i \geq 2 \}\}$$

$y = x$;
$$\{\{ y = x \land x = 2 \times i \land i \geq 2 \}\}$$

$z = (x + y) / 2$;
$$\{\{ z = (x + y) / 2 \land y = x \land x = 2 \times i \land i \geq 2 \}\}$$

$\Rightarrow \{\{ z = (2 \times i + 2 \times i) / 2 \land i \geq 2 \}\}$
Let’s Try One Together (Q4)

Suppose we know that \( i \geq 2 \) at the start, what do we know about \( z \) at the end?

\[
\begin{align*}
\{ \{ i \geq 2 \} \} \\
x &= 2 \times i; \\
\{ \{ x = 2 \times i \land i \geq 2 \} \} \\
y &= x; \\
\{ \{ y = x \land x = 2 \times i \land i \geq 2 \} \} \\
z &= (x + y) / 2; \\
\{ \{ z = (x + y) / 2 \land y = x \land x = 2 \times i \land i \geq 2 \} \} \\
\Rightarrow \{ \{ z = (2 \times i + 2 \times i) / 2 \land i \geq 2 \} \} \\
\Rightarrow \{ \{ z = 2 \times i \land i \geq 2 \} \}
\end{align*}
\]
Let’s Try One Together (Q4)

Suppose we know that $i \geq 2$ at the start, what do we know about $z$ at the end?

$$
\begin{align*}
\{ \{ i \geq 2 \} \} \\
x &= 2 \times i; \\
\{ \{ x = 2 \times i \land i \geq 2 \} \} \\
y &= x; \\
\{ \{ y = x \land x = 2 \times i \land i \geq 2 \} \} \\
z &= (x + y) / 2; \\
\{ \{ z = (x + y) / 2 \land y = x \land x = 2 \times i \land i \geq 2 \} \} \\
\Rightarrow \{ \{ z = (2 \times i + 2 \times i) / 2 \land i \geq 2 \} \} \\
\Rightarrow \{ \{ z = 2 \times i \land i \geq 2 \} \} \\
\Rightarrow \{ \{ z \geq 4 \} \}
\end{align*}
$$
Let’s practice!
(Q5,Q6)
Questions?

• What is the most surprising thing about this?
• What is the most confusing thing?
• What will need a bit more thinking to digest?