Preconditions and proof states for functions (Review)

Pure VS Side-effects over functions

- Mutating variables;
- I/O behaviors;
- Exceptions;

How do we verify the program that is correct?

$$\{Preconditions\}\ Program\ \{Postconditions\}$$

Hoare Logic

```
x = 1
y = x + 1
x = 5
z = x + 5
```

$\rightarrow$

```
y = 1 + 1
x = 5
z = 1 + 5
```

$z \equiv 6$ but we are expecting $z > 8$, so the program is incorrect?

Because of the side effects, program states (variable values) change over the program locations, doing substitution all way down in one step can end up incorrect conclusion about the program.

Hoare Logic

A correct way to guide us how to reason about the code (verify the correctness)?

$$\{Preconditions\}\ Program\ \{Postconditions\}$$
$$\rightarrow$$
$$\{P\}\ \mathcal{P}\ \{Q\}\ \text{(formally, Hoare Logic!!)}$$

Compute Hoare triples along the way

$$\{P\}\ \mathcal{P}\ \{Q\}$$
$$\mathcal{P}\ \text{is a collection of statements, for each statement}\ S$$
$$\rightarrow$$
$$\{P\}\ S\ \{Q\}\ \text{(little hoare triples.)}$$

In this case, for each line of the program execution, we know what is "exactly" to be expected.

Leverage the rules of Hoare Logic

However, computing these hoare triples is not easy!

```
1 // P { x>=1, y>0 }
2     x = 1
3 // Q {x>=1, y>0}
4     y = x + 1
5 // R {y >= 2, x >= 1 }
6     x = 5
7 // Q1 {y >= 2, x = 5}
8     z = x + 5
9 // Q2 {z > 8, y>=2, x=5}
```

Rule Assignment

$$\{[e \mapsto x]P\}\ x = e\ \{P\}$$

e.g. $\{[x = 1]x \geq 1\ i.e.\ 1 \geq 1\}\ x = 1\ \{x \geq 1\}$

Leverage the rules of Hoare Logic

```
1 // P  {x>=1, y>0}
2      x = 1
3 // Q  {x>=1, y>0}
4     y = x + 1
5 // R  {y >= 2, x >= 1 }
6     x = 5
7 // Q1 {y >= 2, x = 5}
8     z = x + 5
9 // Q2 {z > 8, y>=2, x=5}
```

Rule Sequence

$$\{P\}\ s_1\ \{Q\}\ s_2\ \{R\}$$

Propositions are chaining up, how would we compute them?

Weakest preconditions and strongest postcoditions

```
1 //P: {x>=1};  P': {x>-1}
2       x = 1
3       ...
4       z = x + 5
5 //Q: {z > 8}
```

Weakest precondition (WP)

$$\{P'\}\ s\ \{Q\}\ and\ P \rightarrow P' \implies \{P\}\ s\ \{Q\}$$

The precondition that guarantees correctness for the broadest set of inputs.

We want the weakest precondition: the most general precondition needed to establish the postcondition.

The terms "weak" and "strong" refer to how general or specific an assertion is.

$P = \{True\}$, not helpful, it tells nothing about the program.

Weakest preconditions and strongest postcoditions

```
1 //P: {x>=1}
2       ...
3       x = 1
4       z = x + 5
5 //Q': {z = 10}; Q: {z > 8}
```

Strongest postcondition (SP)

$$\{P\}\ s\ \{Q'\}\ and\ Q' \rightarrow Q \implies \{P\}\ s\ \{Q\}$$

When we reason about the program, compute the postcondition (output) as smallest/strongest as possible.

Q & A