

CSE 331 Software Design and Implementation

Homework 1: *Code Reasoning*

Due: Monday April 8 @ 11pm

Contents:

- [Getting Started](#)
- [Turnin Instructions](#)
- [Problems](#)
 - [Problem 1: Forward reasoning with assignment statements](#)
 - [Problem 2: Backward reasoning with assignment statements](#)
 - [Problem 3: Backward reasoning with if/else statements](#)
 - [Problem 4: Wrong weakest precondition for if/else statements](#)
 - [Problem 5: Another wrong weakest precondition for if/else statements](#)
 - [Problem 6: Weakest Conditions](#)
 - [Problem 7: Hoare Triples](#)
 - [Problem 8: Verifying Correctness](#)
 - [Problem 9: Verifying Correctness](#)
 - [Collaboration](#)
 - [Reflection](#)
 - [Time Spent](#)
- [Errata](#)
- [Q & A](#)

Getting Started

You will need to follow a couple of steps to set up for the homework sequence this quarter. Complete these steps by the end of the week so the staff can help you work through any problems.

- a. Run the student-setup script, as explained in the [Tools Overview](#).
- b. Check out your Subversion repository. This quarter, you will receive starter code and submit your assignments through Subversion (SVN).
 - You will learn more about SVN in section. Also have a look at the [Version Control handout](#).
 - Follow the directions in the version control handout (above) to check out your repository. If you are working on your own computer, also see the SVN section of the [Working At Home handout](#).

Turnin Instructions

- Please submit your answers by adding and committing a single PDF file named `hw1_answers.pdf` to the `src/hw1/answers/` directory of your repository. Follow the directions in the version control handout for adding and committing files.
- Your file should be no larger than 3MB. Scanned copies of hand-written documents are fine, as long as they are legible when printed.
- **IMPORTANT!!!** After completing your homework, it is STRONGLY recommended that you run the [validator tool](#). This tool checks for common structural mistakes, such as

naming the file incorrectly or neglecting to add and commit it to your repository correctly (an all-too-common mistake for students new to version control).

- **Submissions that do not follow these guidelines will not be graded.**

Problems

Directions:

- Assume that all numerical values are integers, and that integer overflow will never occur.
- In your answers, you may use any standard symbols for "and" and "or" (& and |, ? and ?, etc.).
- If no precondition is required for a code sequence, simply write {true} to denote the trivial precondition.

Problem 1: Forward reasoning with assignment statements

Find the strongest postcondition of each code sequence by inserting the appropriate assertion in each blank. The first assertion in part (a) is supplied as an example.

a.

```
{ x > 0 }
x = 10;
{ x = 10 }
y = 2 * x;
{ _____ }
z = y + 4;
{ _____ }
x = z / 2;
{ _____ }
y = 0;
{ _____ }
```

b.

```
{ x > 0 }
y = x;
{ _____ }
y = y + 2;
{ _____ }
```

c.

```
{ |x| > 10 }
x = -x;
{ _____ }
x = x / 2;
{ _____ }
x = x + 1;
{ _____ }
```

d.

```
{ y > 2x }
y = y * 2;
{ _____ }
x = x + 1;
{ _____ }
```

Problem 2: Backward reasoning with assignment statements

Find the weakest precondition of each code sequence by inserting the appropriate assertion in each blank.

a.

```
{ _____ }
x = x + 5;
{ _____ }
y = 2 * x;
```

{ y > 10 }

b. { _____ }
 $y = x + 6;$
 { _____ }
 $z = x + y;$
 { z ? 0 }

c. { _____ }
 $y = w - 10;$
 { _____ }
 $x = 2 * x;$
 { x > y }

d. { _____ }
 $t = 2 * s;$
 { _____ }
 $r = w + 4;$
 { _____ }
 $s = 2*s + w;$
 { r > s ? s > t }

Problem 3: Backward reasoning with if/else statements

Find the weakest precondition for the following conditional statement using backward reasoning, inserting the appropriate assertion in each blank. Be sure to explicitly verify that the intermediate postconditions for the two cases imply the total postcondition, i.e. show that $(Q1 \text{ ? } Q2) \text{ ? } Q$.

```
{ _____ }
if (x >= 0)
  { _____ }
  z = x;
  { _____ }
else
  { _____ }
  z = x + 1;
  { _____ }
{ z ? 0 }
```

Problem 4: Wrong weakest precondition for if/else statements

Willy Wazoo says that the CSE 331 formula for $wp(\text{"If } C, S1 \text{ else } S2", Q)$ is too hard to remember. He suggests this simpler definition:

$$wp(\text{"If } C, S1 \text{ else } S2", Q) = wp(S1, Q) \text{ ? } wp(S2, Q)$$

Consider the following Hoare triple:

```
P: { _____ }
if (x == 0)
{
  x = _____; // S1
} else {
  x = _____; // S2
}
Q: { _____ }
```

- Fill in S1, S2 with statements of your choice (but read the rest of this problem before making your choice).
- Fill in Q with a condition of your choice. Compute P according to Willy's definition.
- In the space below, give a concrete initial value of x that satisfies P, and the corresponding final value of x (resulting from executing the code) that does not

satisfy Q. This shows that Willy's definition is incorrect.

Problem 5: Another wrong weakest precondition for if/else statements

Willy accepts your counterexample, but he still believes he can simplify the formula. He proposes to try the equation:

$$\text{wp}(\text{"If } C, S1 \text{ else } S2", Q) = \text{wp}(S1, Q) \text{ ? } \text{wp}(S2, Q)$$

Consider the following Hoare triple:

```
P: { _____ }
if (x == 0)
{
    x = _____; // S1
} else {
    x = _____; // S2
}
Q: { _____ }
```

- Fill in S1, S2 with statements of your choice (but read the rest of this problem before making your choice).
- Fill in Q with a condition of your choice. Compute P according to Willy's definition.
- In the space below, give a concrete initial value of x that does not satisfy P, and the corresponding final value of x (resulting from executing the code) that satisfies Q. This shows that Willy's definition does not give the weakest precondition.

Problem 6: Weakest Conditions

Indicate the weakest condition in each set.

- $\{ x = 20 \}$
 - $\{ x > 10 \}$
 - $\{ x \neq 10 \}$
- (Problem 6b has been retracted.)
- $\{ x > 0 \text{ ? } y > 0 \}$
 - $\{ x > 0 \text{ ? } y > 0 \}$
- $\{ |x+y| > w \}$
 - $\{ x+y > w \}$

Problem 7: Hoare Triples

State whether each Hoare triple is valid. If it is invalid, explain why and show how you would modify the precondition or postcondition to make it valid.

- ```
{ x < 0 }
y = 2 * x;
{ y > 0 }
```

b. 

```
{ x ? y }
z = x - y;
{ z > 0 }
```

c. 

```
{ true }
if (x % 2 == 0)
 y = x;
else
 y = x + 1;
{ y is even }
```

d. 

```
{ x < 0 }
if (x < 100)
 x = -1;
else
 x = 1;
{ x < 0 }
```

## Problem 8: Verifying Correctness

For each block of code, fill in the intermediate assertions, then use them to state whether the precondition is sufficient to guarantee the postcondition. If the precondition is insufficient, explain why and indicate where the assertions don't match up.

Hint: for assignment statements, use backward reasoning to find the weakest precondition that guarantees the postcondition, then see if the given precondition is weaker than the weakest precondition. For if/else statements, you may find a combination of forward and backward reasoning most useful. Follow the rules given in class for what assertion to insert at each point.

a. 

```
{ x > 0 }
y = x - 1;
{ _____ }
z = 2 * y;
{ _____ }
z = z + 1;
{ z > 1 }
```

b. 

```
{ 2x ? w }
y = w - 2;
{ _____ }
x = 2 * x;
{ _____ }
z = x - 2;
{ z ? y }
```

c. 

```
{ y > 0 }
if (x == y)
 { _____ }
 x = -1;
 { _____ }
else
 { _____ }
 x = y - 1;
 { _____ }
{ x < y }
```

## Problem 9: Verifying Correctness

Write a block of code that calculates the smallest even number greater than or equal to  $x$  and stores it in  $y$ . In other words,  $y$  will be assigned either  $x$  or  $x+1$ . Assume  $x$  and  $y$  have already been initialized, and annotate your code with assertions before and after each statement to prove that it is correct. At the end of the block, it should be true that  $y$  is even and that  $y = x$  or  $y = (x+1)$ .

## Collaboration

Please answer the following questions in a file named `collaboration.txt` in your `hw1/answers/` directory.

The standard [collaboration policy](#) applies to this homework.

State whether or not you collaborated with other students. If you did collaborate with other students, state their names and a brief description of how you collaborated.

## Reflection

Please answer the following questions in a file named `reflection.txt` in your `hw1/answers/` directory. Answer briefly, but in enough detail to help you improve your own practice via introspection and to enable the course staff to improve CSE 331 in the future.

- In retrospect, what could you have done better to reduce the time you spent solving this homework?
- What could the CSE 331 staff have done better to improve your learning experience in this homework?
- What do you know now that you wish you had known before beginning the homework?

## Time Spent

Tell us how long you spent on this homework via this catalyst survey:

<https://catalyst.uw.edu/webq/survey/mernst/198071>. (Only include productive time; discount time you worked in your noisy dorm lounge, and don't count time if you were periodically reading email or IMs.)

## Errata

[Problem 6b](#) has been retracted; it didn't have an answer.

## Q & A

This section will list clarifications and answers to common questions about homeworks. We'll try to keep it as up-to-date as possible, so this should be the first place to look (after carefully rereading the homework handout and the specifications) when you have a problem.