Today's Process

- If you haven't completed the solution sheet for Worksheet A, please leave (and go finish it)
- Make sure your student ID (or name) is on your solution sheet
- We'll collect them all, shuffle them, and hand them out – if you get your own, let us know ASAP, since grading your own is not allowed
- Then use a post-it to put your student ID (and name) on the sheet you are grading – otherwise we cannot give you the extra credit you should earn

CSE 331 SOFTWARE DESIGN & IMPLEMENTATION WORKSHEET A

Autumn 2011

---

y≤3 {x=y+1} 2x+y≤11

2

Reminder: logical implication

A⇒B		B: F	B: T
A	F	T	T
	T	F	T

- Apply assignment rule `Q(E) {x = E} Q(x)`
- In this case, directly from the triple
  - `E ≡ y+1`
  - `Q(x) ≡ 2x+y≤11`
- So
  - `Q(E) ≡ 2(y+1)+y≤11`
    `≡ 2y+2+y≤11`
    `≡ 3y≤9`
    `≡ y≤3`
- The precondition is `y≤3`, which implies `Q(E)`, so the triple is **true**

> - 2 points for **true**
> - 0 points for **false**
> - 1 point for **false** with an arithmetic mistake

CSE331 11au

---

n is even ∧ n≤100 {n = n + 2} n is even ∧ n≤101

3

- Apply assignment rule; in this case, directly from the triple
  - `E ≡ n+2`
  - `Q(n) ≡ n is even ∧ n≤101`
- So
  - `Q(E) ≡ (n+2 is even) ∧ n+2≤101`
    `≡ (n+2 is even) ∧ n≤99`
- The precondition is **n is even ∧ n≤100**
  - The precondition conjunct **n is even ⇒ n+2 is even**
  - The precondition conjunct **n≤100** does not ⇒ **n ≤99**
  - So the triple is **false**

> - 2 points for **false**; 0 points for **true**
> - 1 point for **true** with an arithmetic mistake but not for getting the implication between `n≤100` and `n≤99` wrong

---

true {x = 5; y = 0} x=5

4

- Introduce intermediate assertion
  - **true** {x = 5} **x=5** {y = 0} **x=5**
- Work backwards (forwards works, too)
  - **x=5** {y = 0} **x=5**
    is **true** by trivial application of assignment rule
  - **true** {x = 5} **x=5**
    is also **true** by trivial application of assignment rule
- So the triple is **true**

> - 2 points for **true**
> - 0 points for **false**
> - No partial credit

CSE331 11au

---

x=2∧x=3 {x = 5} x=0

5

- Simplify to
  false {x = 5} **x=0**
- A Hoare triple with a **false** precondition is always **true** – the same as **false** precondition allowing a piece of code to provide any behavior
  - Or push the assignment rule through to get
    `false ⇒ 0=5, which also yields true since`
    `false ⇒ false is true`
- So the triple is **true**

> - 2 points for **true**
> - 0 points for **false**
> - No partial credit

CSE331 11au

---

true {System.out.println("Hello world")} false

6

- Because the print statement doesn't change the program state, this reduces to
  true {no-op} **false**
- Does **true** ⇒ **false**? No
- So the triple is **false**

> - 2 points for **false**
> - 0 points for **true**
> - No partial credit

CSE331 11au

false {System.out.println("Hello world")} true

7

- See x=2∧x=3 {x = 5} x=0
- The triple is true

> - 2 points for true
> - 0 points for false
> - No partial credit

x=0 {while x == 0 do x = x+1}x=1

8

- Loops of the form P{while B do S}Q require a loop invariant I to be defined and three properties to
  - A. The invariant holds on entry to the loop: P ⇒ I
  - B. The invariant holds after execution of the loop body: B∧I {S} I
  - C. If the loop terminates, the postcondition can be proved: ¬B∧I ⇒ Q
- Define I as x=0∨x=1
  - A. x=0 ⇒ x=0∨x=1
  - B. x=0∧(x=0∨x=1) {x=x+1} x=0∨x=1
    x=0 {x=x+1} x=0∨x=1

    Q(x) ≡ x=0∨x=1 so Q(E) ≡ x+1=0∨x+1=1
    Q(E) ≡ x=-1∨x=0
    Precondition x=0 ⇒ Q(E)
  - C. x!=0∧(x=0∨x=1) ⇒ x=1
- So the triple is true

> - 2 points for true
> - 1 point for false with correct invariant
> - 0 points for false with incorrect invariant

x=1 {while x != 0 do x = x+1} x=100

9

- Define I as x≤100
  - A. x=1 ⇒ x≤100
  - B. Q(x) ≡ x≤100 so Q(E) ≡ x+1≤100
    Q(E) ≡ x≤99
    Precondition x=1 ⇒ Q(E)
  - C. x=0∧(x≤100) does not ⇒ x=100
- Could there be an I that works? No. Negating the condition for the third part will never let x=0∧I ⇒ x=100 for any I

> - 2 points for false
> - 0 points for true

2x=3y {x = 2x} x=3y

10

- Q(x) ≡ x=3y
- Q(E) ≡ Q(2x)
  ≡ (2x)=3y
  ≡ 2x=3y
- So the precondition 2x=3y ⇒ 2x=3y [Q(E)]
- Yes, so the tuple is true

> - 2 points for true
> - 0 points for false
> - No partial credit

x>2 {if x>2 then y=1 else y=-1} y>0

11

- Conditionals of the form P {if C S_1 else S_2} Q require
  (P∧C {S_1} Q) ∧ (P∧ ¬C {S_2} Q)
- (x>2∧x>2 {y=1} y>0) ∧
  (x>2∧x≤2 {y=-1} y>0)
- The first conjunct is true: x>2 {y=1} y>0
- The second conjunct is true: false {y=-1} y>0
- So the triple is true

> - 2 points for true
> - 0 points for false
> - No partial credit

x/y – remainder (r) and quotient (q)

12

true {

```
r = x;
q = 0;
while (y <= r) {
    r = r - y;
    q = 1 + q;
}
```

x = r + yq ∧ y > r

I ≡ x=r+yq
A. (r=x∧q=0) implies x=r+yq
   (r=x) implies x=r
B. (y≤r)^I {r=r-y;q=q+1} I
   x=(r-y)+y(q+1)
   x=r-y+yq+y
   x=r+yq
C. y>r∧x=r+y×q implies x=r+yq ∧ y>r
QED

> - -2 for reporting that the triple is false
> - -3 points for incorrect loop invariant
> - -1 point for each of the three subparts that is missing
> - -1 or -2 for confusing reasoning

13

```
n > 0 {
  k=0;
  j=1;
  while (k!=n){
    k=k+1;
    j=2*j;
  }
}
j = 2^n
```

I ≡ $j=2^k$
A. $(k=0 \wedge j=1) \Rightarrow j=2^k$
B. $(k!=n) \wedge j=2^k$ {k=k+1;j=2*j} $j=2^k$
  $2j=2^{(k+1)}$
  $2j=2^k 2$
  $j=2^k$
C. $(k=n \wedge j=2^k) \Rightarrow j=2^n$
  QED

- -2 for reporting that the triple is **false**
- -3 points for incorrect loop invariant
- -1 point for each of the three subparts that is missing
- -1 or -2 for confusing reasoning

14

```
n > 0 {
  k=0;
  j=1;
  while (k!=n){
    k=k+1;
    j=2+j;
  }
}
j = 2n+1
```

I ≡ $j=2^k$
A. $(k=0 \wedge j=1) \Rightarrow j=2^k$
B. $(k!=n) \wedge j=2^k$ {k=k+1;j=2*j} $j=2^k$
  $2j=2^{(k+1)}$
  $2j=2^k 2$
  $j=2^k$
C. $(k=n \wedge j=2^k) \Rightarrow j=2^n$
  QED

I ≡ $j=2k+1$
A. $(k=0 \wedge j=1) \Rightarrow j=2k+1$
B. $(k!=n) \wedge j=2k+1$ {k=k+1;j=2+j} $j=2k+1$
  $2+j=2(k+1)+1$
  $2+j=2k+2+1$
  $2+j=2k+3$
  $j=2k+1$
C. $(k=n \wedge j=2^k) \Rightarrow j=2^n$
  QED
QED

- -3 for reporting that the triple is **false**
- -4 for an entirely separate proof from the previous item
- -1 to -3 for confusing association between the previous proof and this proof