CS100J 06 March 2008

Read: Sec. 2.3.8 and chapter 7 on loops. The lectures on the ProgramLive CD can be a big help.

Some anagrams

A decimal point I'm a dot in place
Debit card Bad credit
Dormitory Dirty room
Schoolmaster The classroom
Statue of liberty Built to stay free
The Morse code
Western Union No wire unsent
Parishioners I hire parsons

Animosity Is no amity
Desperation A rope ends it
Funeral Real fun
Slot machines Cash lost in 'em
Snooze alarms Alas! No more Z's
Vacation times I'm not as active
George Bush He bugs Gore
The earthquakes That queen shake

Circumstantial evidence Can ruin a selected victim
Victoria, England's queen Governs a nice quiet land
Eleven plus two Twelve plus one (and they have 13 letters!)

1

Announcements

- 1. Prelim 2 next Thursday evening, 7:30PM, Olin 155. Yes, for-loops are not on this prelim.
- 2. Please complete an online questionnaire concerning your TA.

http://www.engineering.cornell.edu/TAEval/menu.cfm

This is a midterm evaluation. It is important, because your constructive comments are used to help the TA improve, which may help you in this course.

You probably received an email about this. Please complete the survey this week!

2

Assertion: true-false statement placed in a program to assert that it is true at that place.

x = product of 1..n

x ? n 1 x ? n 2

2

Precondition: assertion placed before a segment **Postcondition:** assertion placed after a segment

precondition

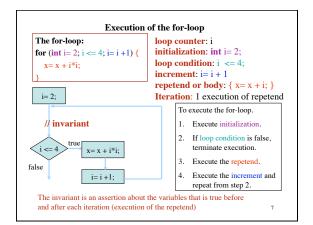
// x = sum of 1..n n = n + 1; x = x + n;// x = sum of 1..npostcondition

4

Solving a problem

| x = x + n; | What statement do you put here so that segment is correct? (if precondition is true, execution of segment should make postcondition true.)

| x = x + n; | precondition | x = x + n; | x = x



```
If k is the next
// Process integers in a..b —
                                                   integer to process,
                                                   which ones have
// invariant: integers in a..k-1 have been processed
                                                   been processed?
for (int k=a; k \le b; k=k+1) {
                                           Command
                                                               A. 0..k
       Process integer k;
                                                to do
                                            something
                                                             B. 0..k-1
                                           and
equivalent
                                                               C. a..k
// post: the integers in a..b have been processed
                                                             D. a..k-1
                                            condition
                                                     E. None of these
Loop invariant says which integers
have been processed (and what that
                                          Iteration: 1 execution of
means). It is true before and after
                                                           repetend
each iteration.
                                            invariant: unchanging
```

```
Finding an invariant: something that is true before
        and after each iteration (execution of the repetend).
// Store in double variable v the sum
                                                    Command to do
1/1 + 1/2 + 1/3 + 1/4 + 1/5 + ... + 1/n
                                                     something and
v=0;
                                                         equivalent
// invariant: v = 1/1 + 1/2 + ... + 1/(k-1)
                                                       postcondition
for (int k=1; k \le n; k=k+1) {
       Process k
                                 What is the invariant?
// v = 1/1 + 1/2 + ... + 1/n
                                 A. v = 1/1 + 1/2 + ... + 1/n
                                 B. v = 1/0 + 1/1 + ... + 1/k
v = sum of 1/i for i in
                                 C. v = 1/0 + 1/1 + ... + 1/(k-1)
    range 1..k-1
                                 D. v = 1/1 + 1/1 + ... + 1/(k-1)
                                 E. None of these
```

```
Find invariant: true before and after each iteration
// set x to no. of adjacent equal pairs in s[0..s.length()-1] - Command
// invariant: x = no. of adjacent equal pairs in s[0..k-1]
                                                                 something
                                                                      and
                                                                 equivalent
for (int k = 0; k < s.length(); k = k + 1) {
                                                                  post-
condition
        Process k
                                     for s = 'ebeee',
                                     x = 2.
// x = \text{no. of adjacent equal pairs in s}[0..s.length()-1]
k: next integer to process. What is the invariant?
Which ones have been
                            A. x = no. adj. equal pairs in s[1..k]
processed?
                            B. x = no. adj. equal pairs in s[0..k]
A. 0..k
              C. a..k
                            C. x = no. adj. equal pairs in s[1..k-1]
B. 0..k-1
             D. a..k-1
                            D. x = no. adj. equal pairs in s[0..k-1]
E. None of these
                            E. None of these
```

```
1. What is the invariant?
  Being careful
                                                              Command
// { String s has at least 1 char }
// Set c to largest char in String s
                                                           postcondition
                                               2. How do we initialize c
// inv: c is largest char in s[0..k-1]
                                                   and k?
for (int k = ; k < s.length(); k = k + 1) {
                                               A. k=0; c= s.charAt[0];
    // Process k:
                                               B. k= 1; c= s.charAt[0];
                                               C. k= 1; c= s.charAt[1];
                                               D. k=0; c=s.charAt[1];
// c = largest char in s[0..s.length()-1]
                                               E. None of the above
An empty set of characters or integers has no maximum. Therefore,
be sure that 0..k-1 is not empty. Therefore, start with k = 1.
```

```
Methodology for developing a for-loop

1. Recognize that a range of integers b..c has to be processed

2. Write the command and equivalent postcondition.

3. Write the basic part of the for-loop.

4. Write loop invariant, based on the postcondition.

5. Figure out any initialization.

6. Implement the repetend (Process k).

// Process b..c

Initialize variables (if necessary) to make invariant true.

// Invariant: range b..k-1 has been processed

for (int k= b; k <= c; k= k+1) {

// Process k

}

// Postcondition: range b..c has been processed
```