

CS100J 16 Oct, 2007
Assignment A5: loops

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	Animosity	Is no amity
Debit card	Bad credit	Desperation	A rope ends it
Dormitory	Dirty room	Funeral	Real fun
Schoolmaster	The classroom	Slot machines	Cash lost in 'em
Statue of liberty	Built to stay free	Snooze alarms	Alas! No more Z's
The Morse code	Here come dots	Vacation times	I'm not as active
Western Union	No wire unsent	George Bush	He bugs Gore
Parishioners	I hire parsons	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!)

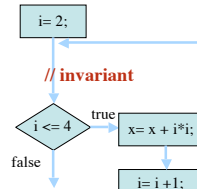
1

Execution of the for-loop

The for-loop:

```
for (int i=2; i <= 4; i= i+1) {
    x = x + i*i;
}
```

loop counter: i
initialization: int i=2;
loop condition: i <= 4;
increment: i= i+1
repetend or body: { x= x + i; }
Iteration: 1 execution of repetend



The invariant is an assertion about the variables that is true before and after each iteration (execution of the repetend)

2

// Process integers in a..b

// invariant: integers in a..k-1 have been processed

```
for (int k= a; k <= b; k= k + 1) {
    Process integer k;
}
```

// post: the integers in a..b have been processed

Loop invariant says which integers have been processed (and what that means). It is true before and after each iteration.

If k is the next integer to process, which ones have been processed?

Command to do something and equivalent post-condition

A. 0..k
 B. 0..k-1
 C. a..k
 D. a..k-1

E. None of these

Iteration: 1 execution of repetend
invariant: unchanging

3

Finding an invariant: something that is true before and after each iteration (execution of the repetend).

// Store in double variable v the sum

// 1/1 + 1/2 + 1/3 + 1/4 + 1/5 + ... + 1/n

// invariant: v = 1/1 + 1/2 + ... + 1/(k-1)

```
for (int k= 1; k <= n; k= k+1) {
    Process k
}
```

// v = 1/1 + 1/2 + ... + 1/n

What is the invariant?

A. v = 1/1 + 1/2 + ... + 1/n
 B. v = 1/0 + 1/1 + ... + 1/k
 C. v = 1/0 + 1/1 + ... + 1/(k-1)
 D. v = 1/1 + 1/1 + ... + 1/(k-1)
 E. None of these

4

Find invariant: true before and after each iteration

// set x to no. of adjacent equal pairs in s[0..s.length()-1]

// invariant: x = no. of adjacent equal pairs in s[0..k-1]

```
for (int k= 0; k < s.length(); k= k + 1) {
    Process k
}
```

// x = no. of adjacent equal pairs in s[0..s.length()-1]

k: next integer to process.
 Which ones have been processed?

A. 0..k
 B. 0..k-1
 C. a..k
 D. a..k-1
 E. None of these

What is the invariant?

A. x = no. adj. equal pairs in s[1..k]
 B. x = no. adj. equal pairs in s[0..k]
 C. x = no. adj. equal pairs in s[1..k-1]
 D. x = no. adj. equal pairs in s[0..k-1]
 E. None of these

5

Being careful

1. What is the invariant?

// { String s has at least 1 char }

// Set c to largest char in String s

// inv: c is largest char in s[0..k-1]

```
for (int k= ; k < s.length(); k= k + 1) {
    // Process k;
}
```

// c = largest char in s[0..s.length()-1]

2. How do we initialize c and k?

A. k=0; c= s.charAt[0];
 B. k= 1; c= s.charAt[0];
 C. k= 1; c= s.charAt[1];
 D. k= 0; c= s.charAt[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 <= b; k= k+1) {
    // Process k
}
// Postcondition: range b..c has been processed
```

7

The while loop syntax and semantics

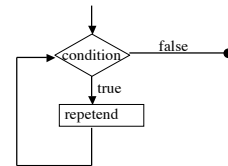
while (<condition>)
 <repetend>

<condition>: a boolean expression.

<repetend>: a statement.

while (<condition> {
 sequence of declarations
 and statements
}

BUT: We always make the <repetend> a block.



8

for-loop and equivalent while-loop

```
// Sum the squares of 5..60
int x= 0;
// inv: x = sum of squares of 5..k-1
for (int k= 5; k <= 60; k= k+1) {
    // Process k
    x= x + k*k;
}
// x = sum of squares of 5..60

// Sum the squares of 5..60
int x= 0;
int k= 5;
// inv: x = sum of squares of 5..k-1
while (k <= 60) {
    // Process k
    x= x + k*k;

    k= k + 1;
}
// x = sum of squares of 5..60
```

9