Announcements

- Prelim 2 conflicts due by midnight tonight
- Lab 11 is out
  * Due in 2 weeks because of Prelim 2
- Review Prelim 2 announcements from previous lecture
- A4 is due Thursday at midnight
  * Can look at webpage for redistributed weights

Recall: Important Terminology

- **assertion**: true-false statement placed in a program to assert that it is true at that point
  * Can either be a comment, or an assert command
- **invariant**: assertion supposed to "always" be true
  * If temporarily invalidated, must make it true again
  * Example: class invariants and class methods
- **loop invariant**: assertion supposed to be true before and after each iteration of the loop
- **iteration of a loop**: one execution of its body

Preconditions & Postconditions

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

Relationship Between Two
If precondition is true, then postcondition will be true

Solving a Problem

What statement do you put here to make the postcondition true?
A: x = x + 1
B: x = x + n
C: x = x + n+1
D: None of the above
E: I don’t know

Invariants: Assertions That Do Not Change

- **Loop Invariant**: an assertion that is true before and after each iteration (execution of repetend)
  x = 0; i = 2
  while i <= 5:
    x = x + i*i
    i = i + 1
  # x = sum of squares of 2..i-1
  Invariant:
  x = sum of squares of 2..i-1
  in terms of the range of integers that have been processed so far
  The loop processes the range 2..5

- **Invariant**: Assertions That Do Not Change
  x = 0; i = 2
  # Inv: x = sum of squares of 2..i-1
  while i <= 5:
    x = x + i*i
    i = i + 1
  # Post: x = sum of squares of 2..5
  Integers that have been processed: 2, 3, 4, 5
  Range 2..i-1: 2..5
  Invariant was always true just before test of loop condition. So it’s true when loop terminates
  The loop processes the range 2..5
Designing Integer while-loops

```markdown
# Process integers in a..b
# inv: integers in a..k-1 have been processed
k = a
while k <= b:
    process integer k
    k = k + 1
# post: integers in a..b have been processed
```

**Finding an Invariant**

```markdown
# Make b True if n is prime, False otherwise
b = True
k = 2
# invariant: b is True if no int in 2..k-1 divides n, False otherwise
while k < n:
    # Process k:
    if n % k == 0:
        b = False
    k = k + 1
# b is True if no int in 2..n-1 divides n, False otherwise
```

What is the invariant? 1 2 3 ... k-1 k k+1 ... n

**Finding an Invariant**

```markdown
# set x to # adjacent equal pairs in s
x = 0
# inv: x = # adjacent equal pairs in s[0..k-1]
while k < len(s):
    # Process k
    k = k + 1
# x = # adjacent equal pairs in s[0..len(s)-1]
```

k: next integer to process.
What is initialization for k?
A: k = 0
B: k = 1
C: k = -1
D: I don’t know
E: I don’t know

**Reason carefully about initialization**

```markdown
# s is a string; len(s) >= 1
# Set c to largest element in s
# c = ??
while k < len(s):
    # Process k
    k = k + 1
# c = largest char in s[0..len(s)-1]
```