

Introduction

Readings for Friday: *Programming Into Java*, chapters 1, 3. *On to Java*, chapters 21, 28.

Register! Be sure to *get an account* and register (as in the first lab) by Friday, even if you are on the waiting list or seeking concurrent enrollment. Otherwise, you will be dropped!

Readers available at Vick Copy, as indicated in the first handout. (Why were people going to Copy Central??)

Today. More examples of Java. Tracing programs and the use of comments.

Printing results

A Scheme programmer might write this:

```
/** Print all primes in the range LOW .. HIGH,  
 * inclusive, on the standard output. */  
static void print (int low, int high) {  
    if (low <= high) {  
        if (isPrime (low)) // (corrected)  
            System.out.print (low + ", ");  
        print (low+1, high);  
    }  
}
```

Tracing the Code

Understand print by *tracing*. E.g., `print(3, 5)`

- Call assigns `low=3, high=5`.
- Body has form 'if (`low <= high`) S '. Since $3 \leq 5$, do S :
 - S is a *block* of two statements, S_1, S_2 .
 - S_1 : `if (isPrime (low)) S_3`
 - What to do about `isPrime`? Use the *comment*: since 3 is prime, result is true.
 - So now do S_3 (built-in `print`).
 - S_2 : `print(low+1, high);`
 - Again, use the comment: this prints all primes from 4 to 5.
- Lesson: Comments aid understanding. Make them *count!*

Looping

It is more customary to write

```
/** Print all primes in the range LOW .. HIGH,  
 * inclusive, on the standard output. */  
static void print (int low, int high) {  
    for (int i = low; i <= high; i += 1)  
        if (isPrime (i))  
            System.out.print (i + ", ");  
}
```

- 'int i' introduces new local variable.
- for iterates.
- + on Strings means "concatenate"
- print VS. println.

Revisiting Prime Testing

Could also re-write isPrime:

```
/** True iff X is prime. */
static boolean isPrime (int x) {
    if (x == 2)
        return true;
    if (x <= 1 || x % 2 == 0)
        return false;
    int limit = 1 + (int) Math.sqrt (x);
    for (int k = 3; k <= limit; k += 2) // ??
        if (x % k == 0)
            return false;
    return true;
}
```

Main program

```
/** Print all primes in the range given by ARGS
 * on the standard output (ARGS consists of
 * 1 or 2 numerals: an upper bound or lower
 * and upper bounds). */
static void main (String[] args) {
    try {
        if (args.length == 1)
            print (2, Integer.parseInt (args[0]));
        else if (args.length == 2)
            print (Integer.parseInt (args[0]),
                Integer.parseInt (args[1]));
        System.out.println ();
    } catch (NumberFormatException e) {
        System.err.println ("Error: improper number.");
        Usage ();
    }
}
```

- A string of digits is not an int.
- Exceptions for signalling errors.

Lists

- In Scheme, had built-in lists.
- `cons` *constructs* lists; `car` and `cdr` take apart.
- Type is not primitive in Java, but can define it:

```
class ScmList {
    // Instance variables
    private int head;
    private ScmList tail;

    // Constructor (like cons)
    ScmList (int head, ScmList tail) {
        this.head = head; this.tail = tail;
    }

    int car () { return head; }
    ScmList cdr () { return tail; }
    void setCar (int x) { head = x; }
    void setCdr (ScmList x) { tail = x; }
}
```

Creating a Modified List

```
;; SCHEME
;; List of all items in L incremented by n.
(define (incr-list L n)
  (cond ((null? L) '())
        (else (cons (+ n (car L))
                     (incr-list (cdr L) n)))))

// JAVA
/** List of all items in L incremented by n. */
public static ScmList incrList (ScmList L, int n) {
  if (L == null)
    return null;
  else return new ScmList (L.car()+n,
                          incrList(L.cdr(), n));
}
```