Different Points of View

CS 1025 Computer Science Fundamentals I

Stephen M. Watt
University of Western Ontario

Many points of view

- Before we dive into object-oriented programming, let's look at a couple of other points of view.
- We will take a quick look at a couple of other ways of programming and few different development environments.
- Don't worry about getting every detail at this stage, this is more just to get an impression.

Imperative Programming

- This is a C program to compute factorials, $n! = n \times (n - 1) \times \ldots \times 3 \times 2 \times 1$.

```
int factorial(int n) {
    int i, prod;
    prod = 1;
    for (i = 1; i <= n; i++)
        prod = prod * i;
    return prod;
}
```

- The main things to notice are the assignments and the loop.

Functional Programming

- This is a Scheme program to compute n!

```
(define factorial (lambda (n)
        (if (= n 1)
            1
            (* n (factorial (- n 1))) ) ))
```

- Scheme is a member of the Lisp family of languages, the first of which from about 1960.
- It *uniformly* uses the syntax **(operator arg1 arg2 ...)**.

- This program uses *recursion* and the fact that, for n > 1, n! = n × (n – 1)!

Uniformity *vs* Convenience

- Which is better, to have convenient, but irregular syntax, like `1 + cos(theta)`,

  or to have a completely uniform, but less convenient syntax, like `(+ 1 (cos theta))`.

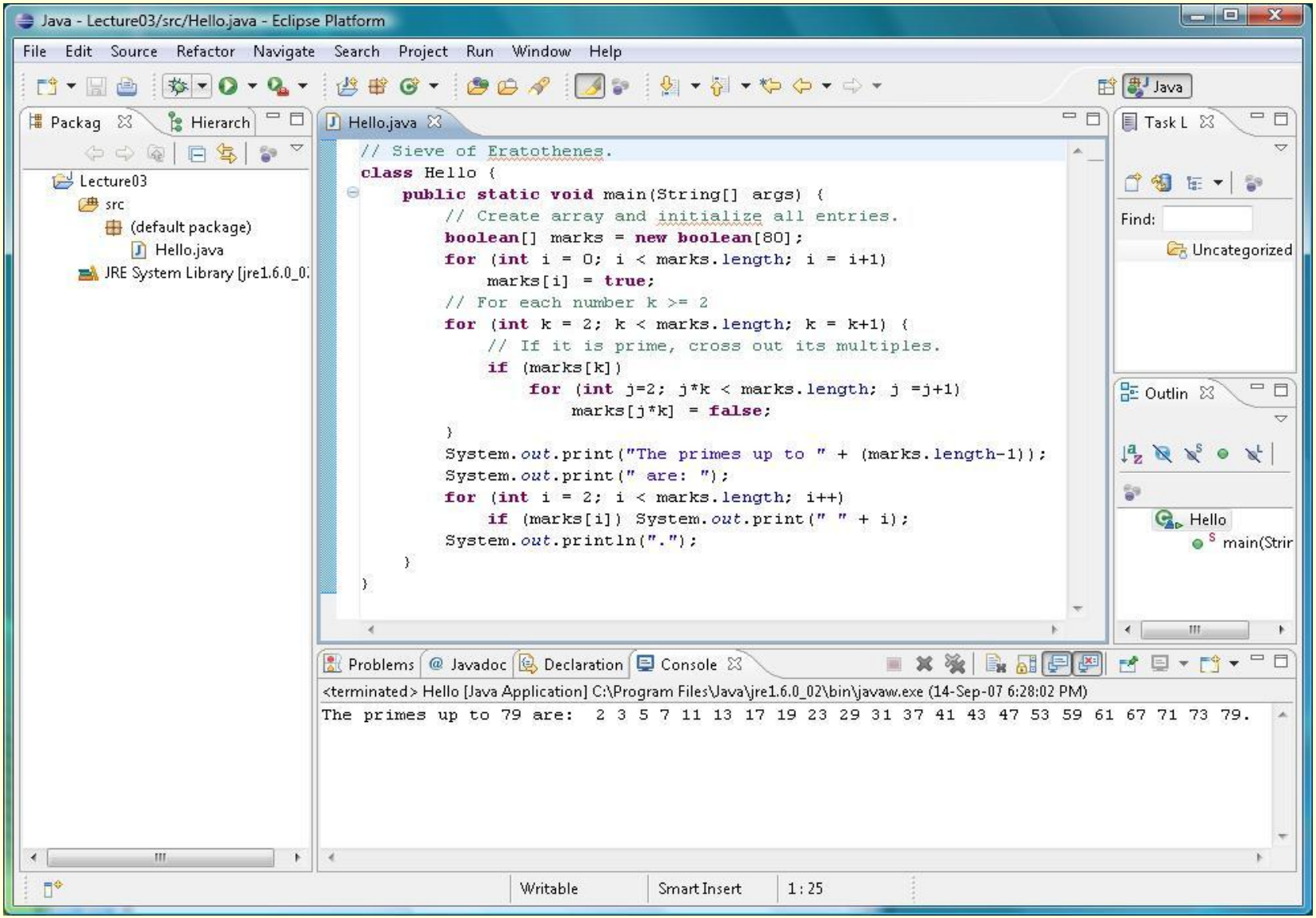
- The irregular syntax is easier to use and the regular syntax is easier to compose.

- For single jobs a Swiss Army knife is better than a Lego brick, but *how would you make a replica of the Empire State building out of Swiss Army knives.*

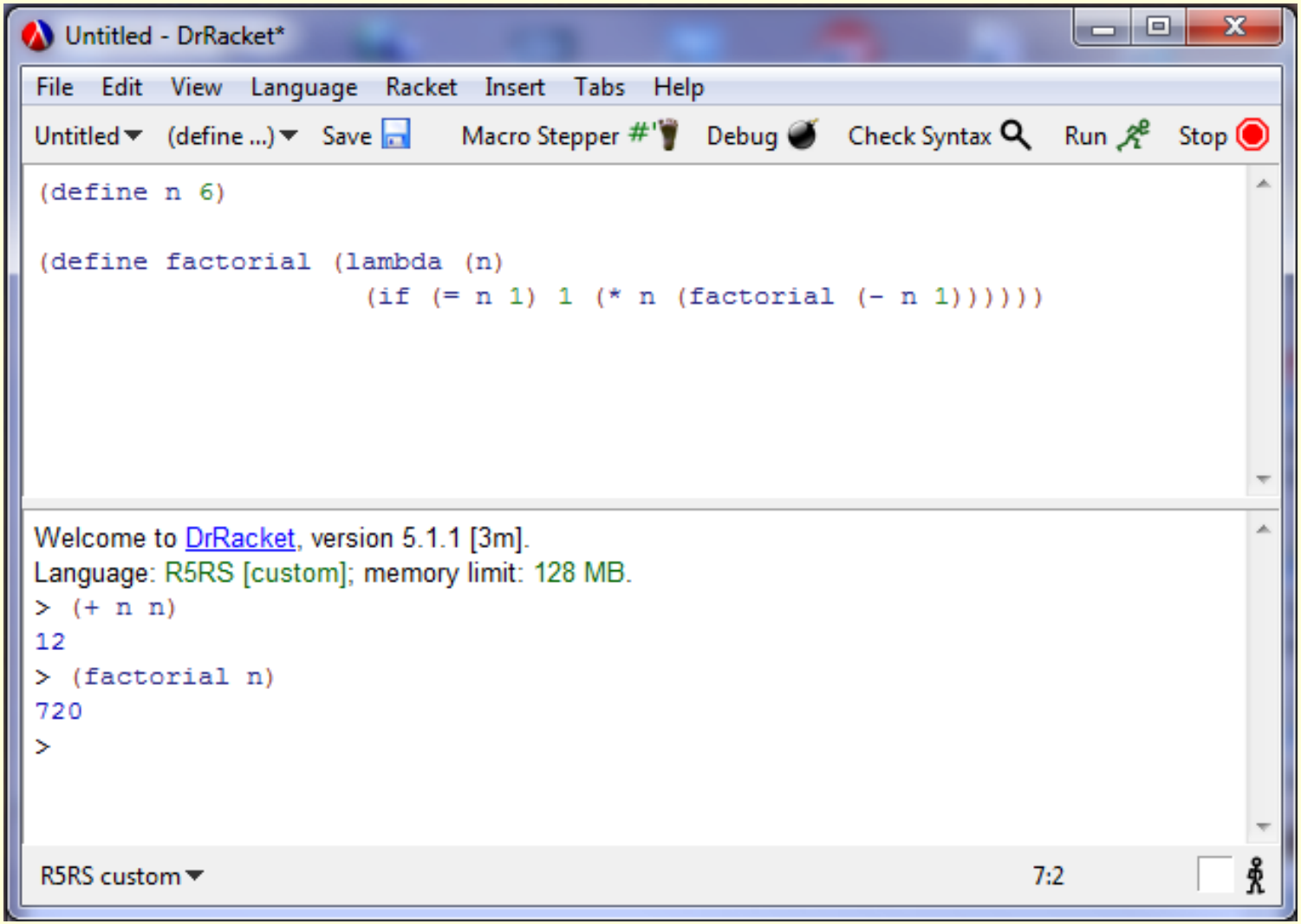
The Lisp family of languages

- Most things are made up of “cons” cells that contain two things.
- The things they contain can be values or pointers to other things.
- Can make complicated data structures from these.

The *Eclipse* IDE for Java Development



The *DrRacket* IDE for Scheme



The *Maple* Environment