

Programming Paradigms

Reading: *Chapters 10, 11, 12*
(First pass only)

Programming Paradigms

- A programming paradigm is defined by the kinds of data and procedural abstractions it supports
- Major Programming Paradigms
 - ⇒ Imperative programming languages (**C**)
 - ⇒ Functional programming languages (**Lisp**)
 - ⇒ Object-oriented programming languages (**Smalltalk**)
 - ⇒ Logic programming languages (**Prolog**)
- Multiparadigm programming languages (**Java**)

Imperative Languages

- Defining features
 - ⇒ Variables
 - ⇒ Assignment
 - ⇒ Sequential execution
- Oldest paradigm
- Imperative programs tell a computer what to do at each step
- Most similar to the underlying hardware
- Close connection between
 - ⇒ variables and memory locations
 - ⇒ operations and machine instructions

Imperative Example: C

```
int fact(int n)
{
    int result;

    result = 1;
    while (n > 0) {
        result = result * n;
        n = n - 1;
    }
    return result;
}
```

result	n
1	10

Imperative Example: C

```
int fact(int n)
{
    int result;
    result = 1;
    while (n > 0) {
        result = result * n;
        n = n - 1;
    }
    return result;
}
```

=

result	n
1	10
10	9

Imperative Example: C

```
int fact(int n)
{
    int result;
    result = 1;
    while (n > 0) {
        result = result * n;
        n = n - 1;
    }
    return result;
}
```

	result	n
	1	10
=	10	9
=	90	8

Imperative Example: C

```
int fact(int n)
{
    int result;
    result = 1;
    while (n > 0) {
        result = result * n;
        n = n - 1;
    }
    return result;
}
```

	result		n
	1	X	10
=	10	X	9
=	90	X	8
=	720		7

Imperative Example: C

```
int fact(int n)
{
    int result;

    result = 1;
    while (n > 0) {
        result = result * n;
        n = n - 1;
    }
    return result;
}
```

	result	n
	1	10
=	10	9
=	90	8
=	720	7
	5040	6
	30240	5
	151200	4
	604800	3
	1814400	2
	3628800	1