Iterative algorithms

- Looping constructs (e.g. while or for loops) lead naturally to **iterative** algorithms
- Can conceptualize as capturing computation in a set of “state variables” which update on each iteration through the loop

Iterative multiplication by successive additions

- Imagine we want to perform multiplication by successive additions:
  - To multiply a by b, add a to itself b times
- State variables:
  - i – iteration number; starts at b
  - result – current value of computation; starts at 0
- Update rules
  - i ← i -1; stop when 0
  - result ← result + a

```python
def iterMul(a, b):
    result = 0
    while b > 0:
        result += a
        b -= 1
    return result
```