Stack

• Allow access to only the top element
• Operations supported in stack:
  – Add element at top of a stack (push)
  – Extract element from the stack (pop)
  – Inspect the top element of the stack (peek)
• All operations take a constant amount of time.
Stack Application

• Compiler checks whether symbols are balanced
  – E.g., { has }
  – Counting the number of each symbol is not sufficient
    • E.g., { ( } ) is not legal.
Stack Application (cont.)

1. Make an empty stack
2. Read all symbols from input
   2.1 If it is an opening symbol, push to stack
   2.2 If it is a closing symbol
      a) If stack is empty report error
      b) If stack is not empty and the closing symbol does not
         match the opening symbol, report an error.
3. After reading symbols, if the stack is not empty report an error.
Queue

• Allow access the least recently (oldest) item
• Printers

• Operations supported in queue:
  – Add element at the back of a queue (add)
  – Extract the front element from the queue (remove or poll)
  – Inspect the front element of the queue (element or offer)

• All operations take a constant amount of time.