Department of
Computer Science

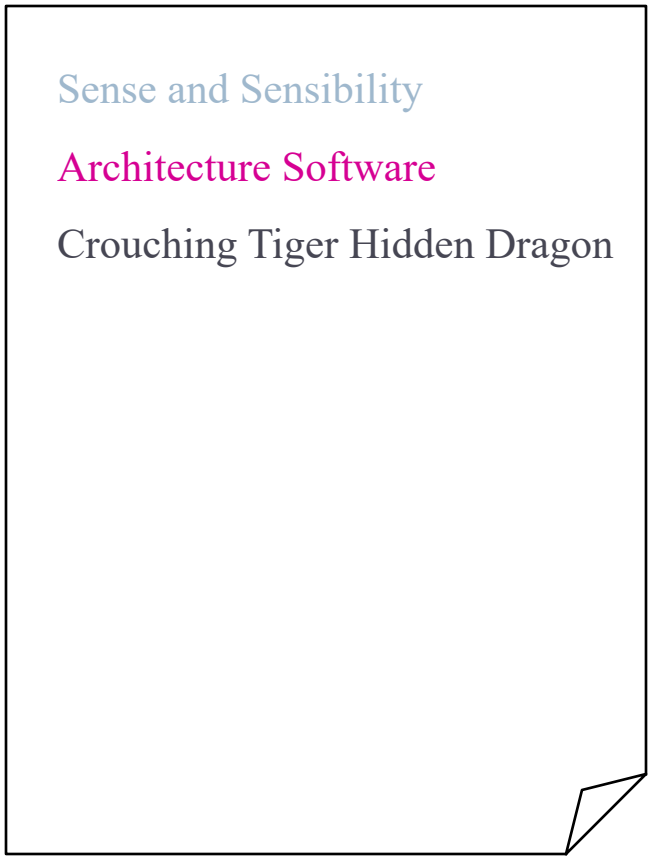
Software Fundamentals

“On the Criteria to Be Used in Decomposing Systems into Modules”

A Canonical Example

Key Word in Context [Parnas72]

1. Input



A Canonical Example

Key Word in Context [Parnas72]

1. Input
2. Circular Shift

A Canonical Example

Key Word in Context [Parnas72]

1. Input
2. Circular Shift
3. Alphabetizing

and Sensibility Sense
Architecture Software
Crouching Tiger Hidden Dragon
Dragon Crouching Tiger Hidden
Hidden Dragon Crouching Tiger
Sense and Sensibility
Sensibility Sense and
Software Architecture
Tiger Hidden Dragon Crouching

A Canonical Example

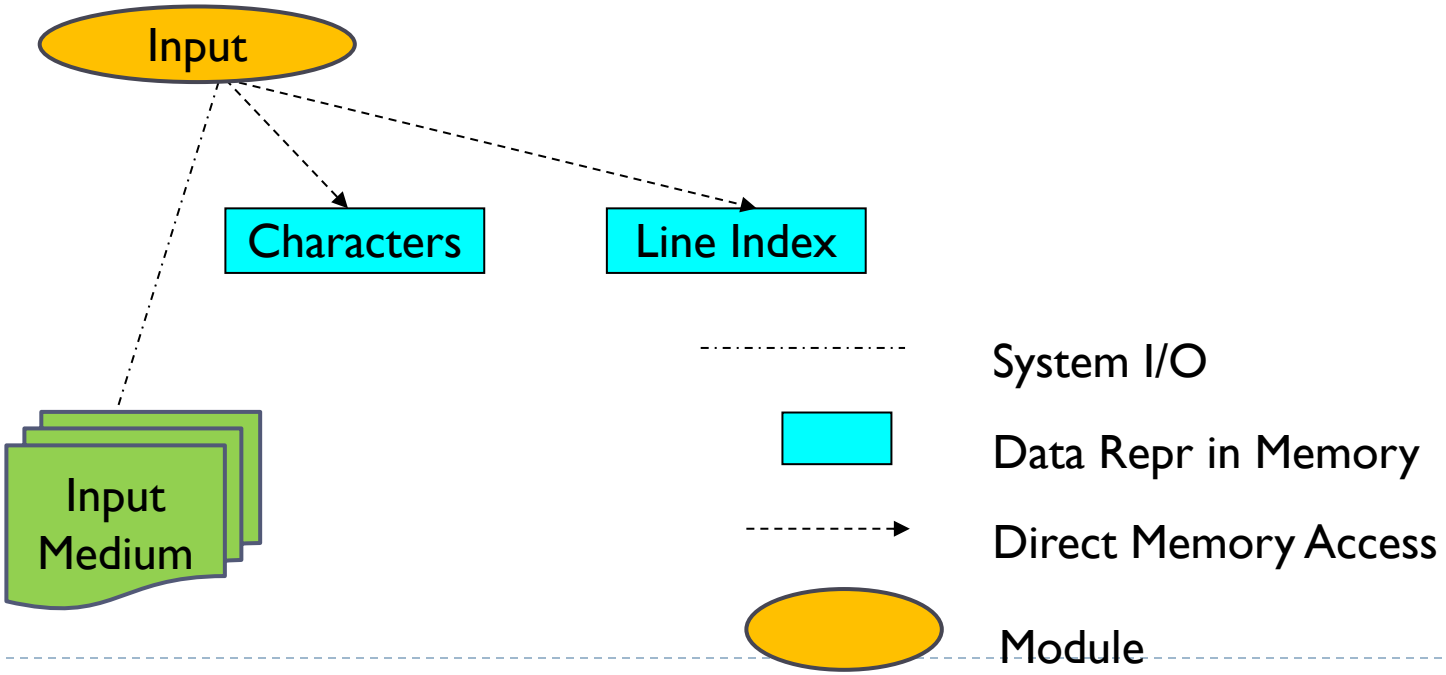
Key Word in Context [Parnas72]

1. Input
2. Circular Shift
3. Alphabetizing
4. Output

Modularization 1

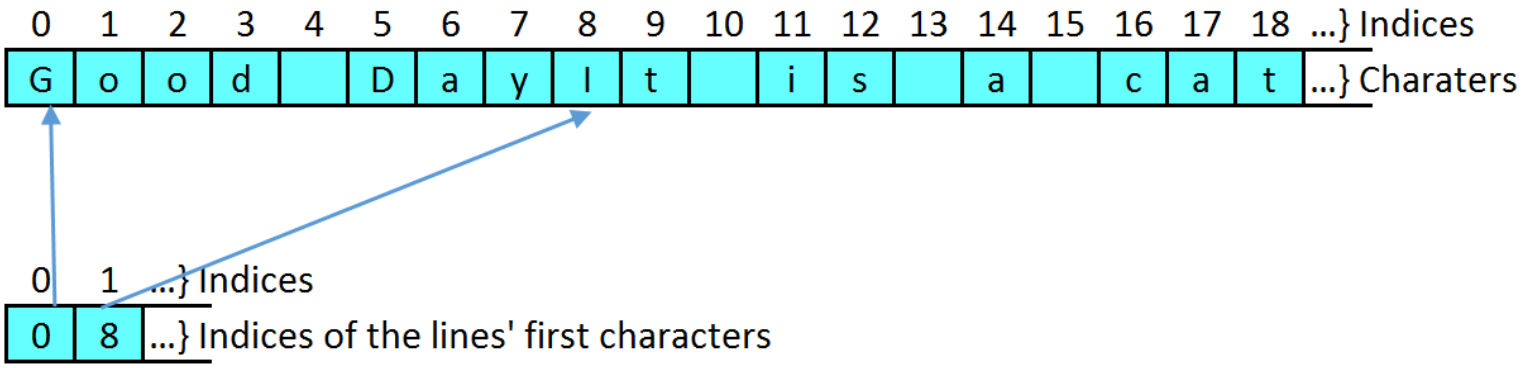
Module1: Input

"*This module reads the data lines from the input medium and stores them in core for processing by the remaining modules. The characters are packed four to a word, and an otherwise unused character is used to indicate the end of a word. An index is kept to show the starting address of each line.*"



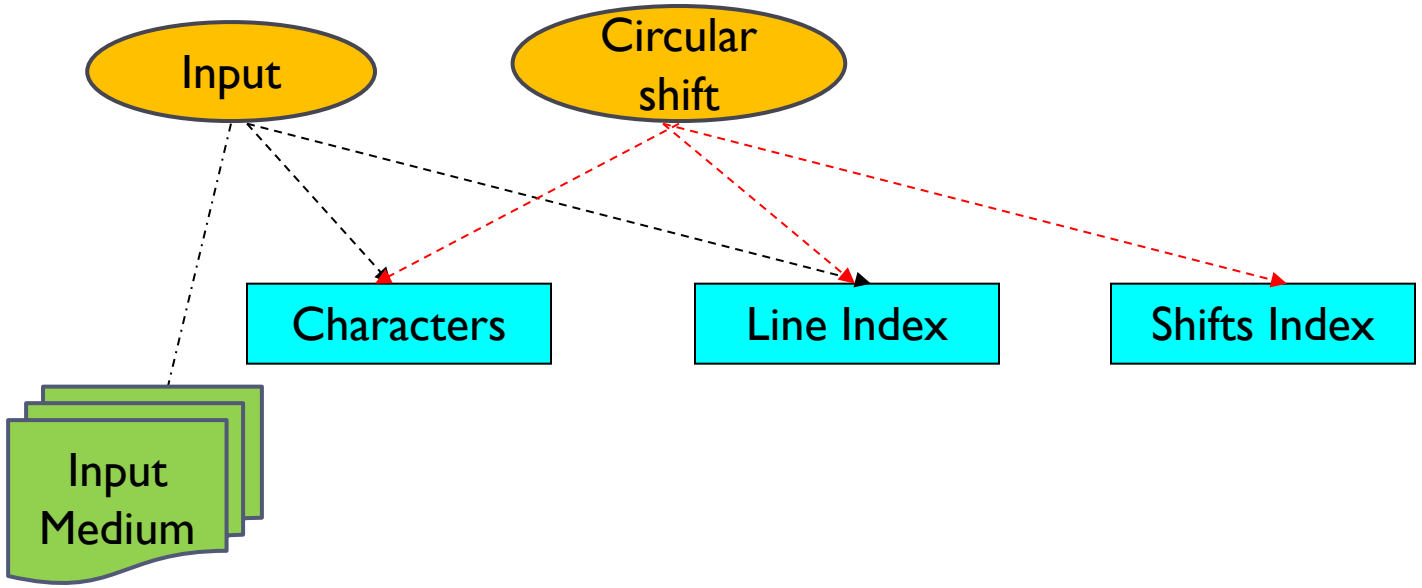
Module1: Input

"*This module reads the data lines from the input medium and stores them in core for processing by the remaining modules. The characters are packed four to a word, and an otherwise unused character is used to indicate the end of a word. An index is kept to show the starting address of each line.*"



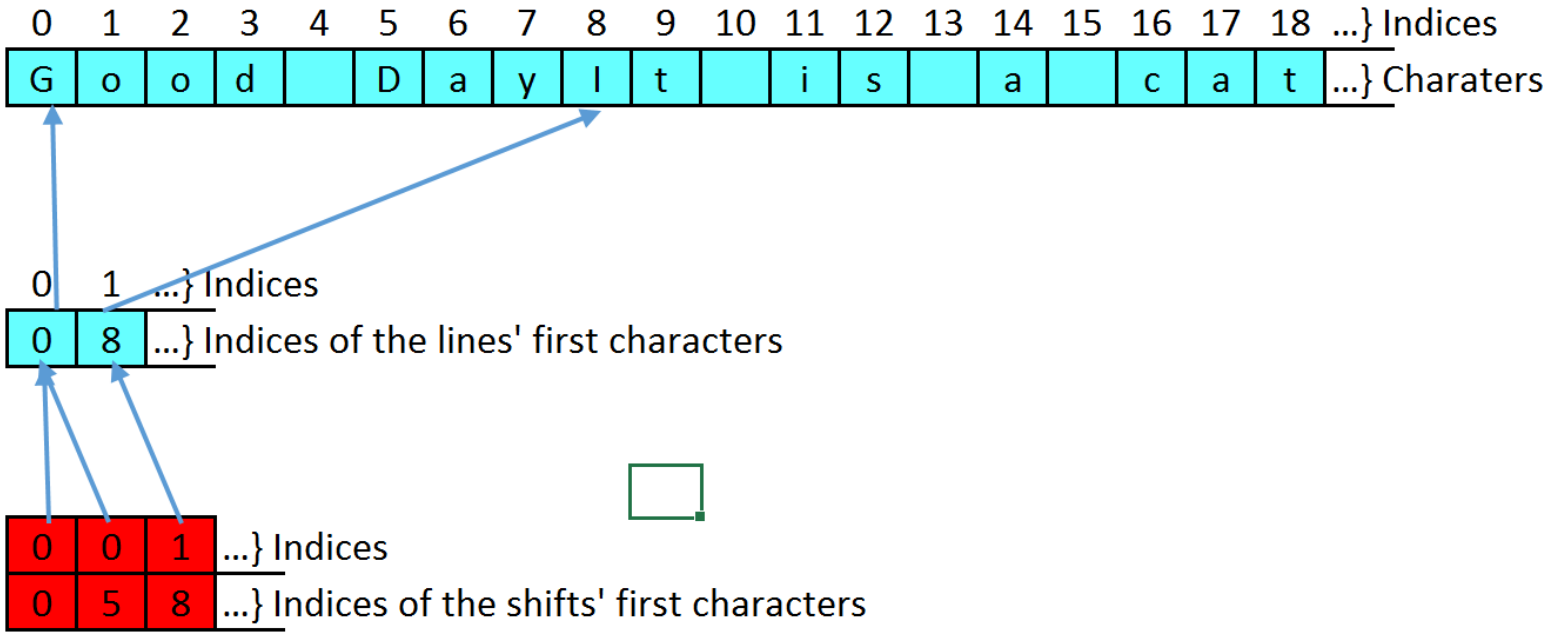
Module 2: Circular Shift

"*This module is called after the input module has completed its work. It prepares an index which gives the address of the first character of each circular shift, and the original index of the line in the array made up by module 1. It leaves its output in core with words in pairs (original line number, starting address)* "



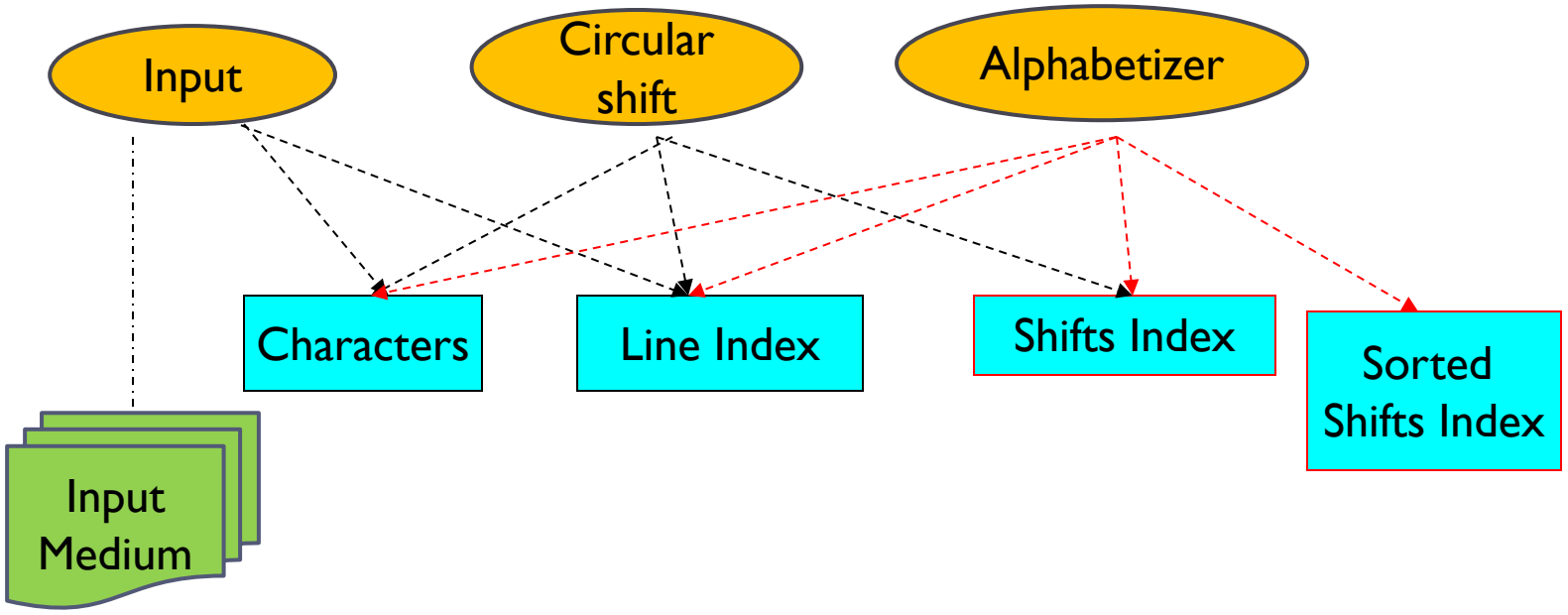
Module 2: Circular Shift

"*This module is called after the input module has completed its work. It prepares an index which gives the address of the first character of each circular shift, and the original index of the line in the array made up by module 1. It leaves its output in core with words in pairs (original line number, starting address)* "



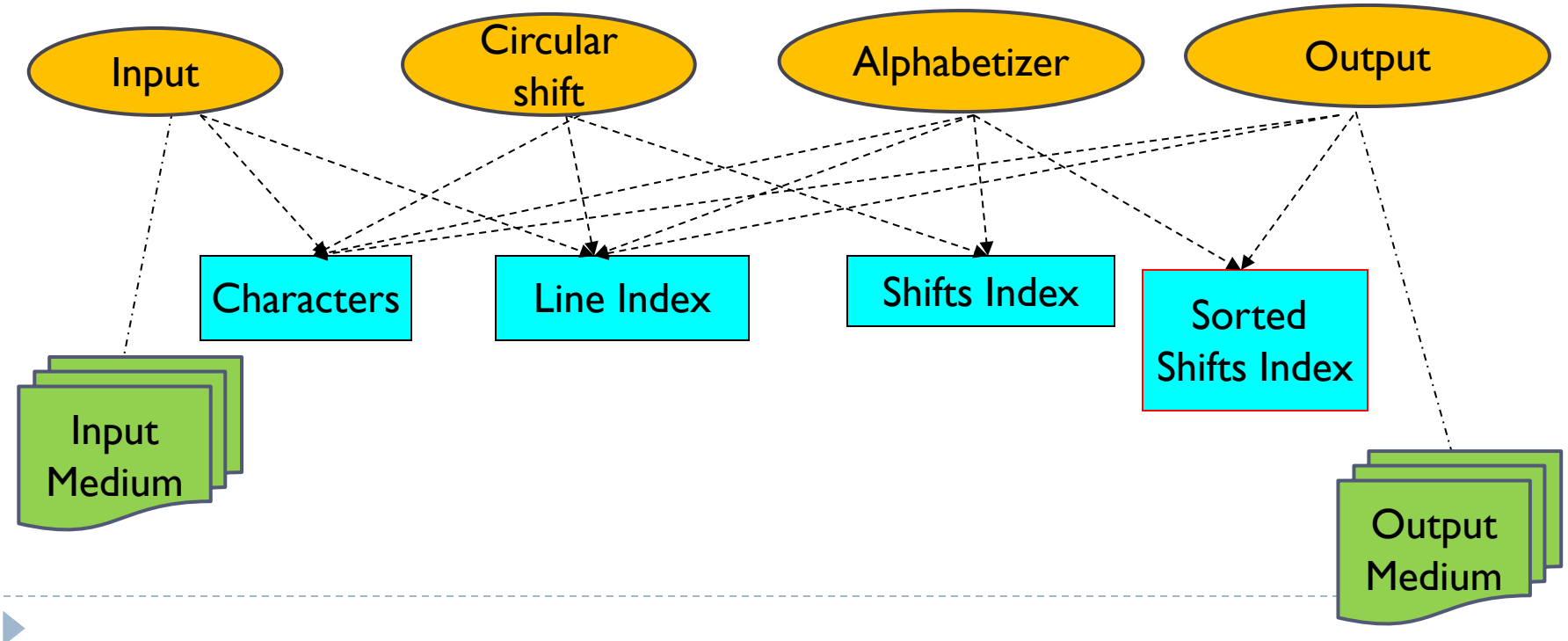
Module 3: Alphabetizing

"*This module takes as input the arrays produced by modules 1 and 2. It produces an array in the same format as that produced by module 2. In this case, however, the circular shifts are listed in another order (alphabetically).* "



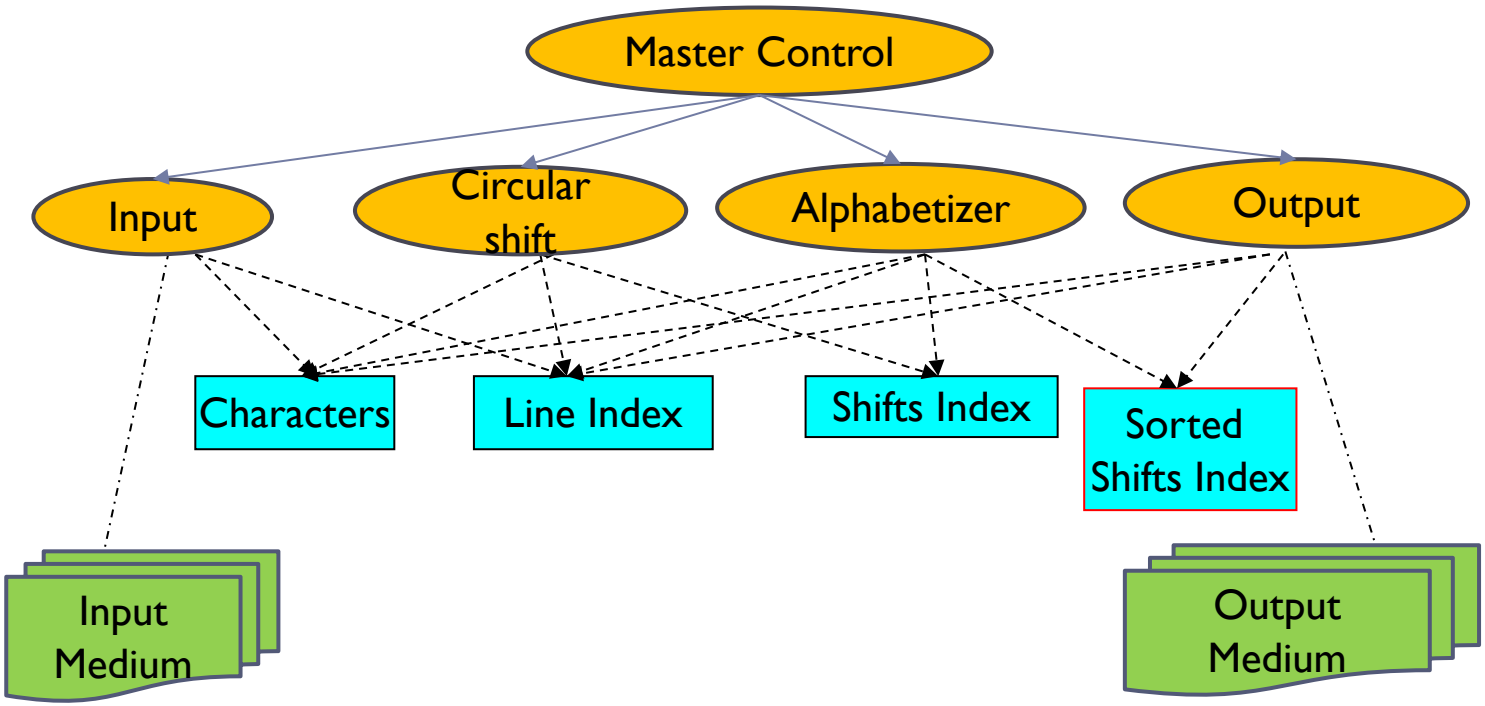
Module 4: Output

"*Using the arrays produced by module 3 and module 1, this module produces a nicely formatted output listing all of the circular shifts. In a sophisticated system the actual start of each line will be marked, pointers to further information may be inserted, and the start of the circular shift may actually not be the first word in the line, etc.* "



Module 5: Master Control

"*This module does little more than control the sequencing among the other four modules. It may also handle error messages, space allocation, etc . "*



Modularization 2

Modularization 2

- **Module 1: Line Storage**
- Module 2: Input
- Module 3: Circular Shift
- Module 4: Alphabetizer
- Module 5: Output
- Module 6: Master Control

Modularization 2