Format for a **Class:**

{

Variables : Formula that holds for the variables.

}

-o-

Notes:

- y is a set if and only if there is z such that y is an element of z; otherwise y is a proper class.
- The statement above implies that every set is a class.

Format for a **Finite Set:**

{

element_1, element_2, …, element_n.

}

-o-

Format for a **Data Structure:**

(

Class Name,

Attribute Domains in order,

Attributes in order,

Interpretation_function_1, Interpretation_function_2, …, Interpretation_function_m,

Functionality_1, Functionality_2, …, Functionality_k,

Documentation.

)

-o-

Notes:

- Usually one interpretation_function is enough, however, if one tries to find patterns, multiple choices might be useful ☺
- Functionality of a class with n attributes is a function that has at most n parameters.
    - The statement above is verified

Format for a **Process:**

(

- Name
- Scoping
    - Key objectives
    - Stakeholders
    - Timeline
    - Priority compared to other objectives
- Defining boundaries
- Input Output
    - Resources to complete a process
    - Goal
- Steps
    - Steps defining an execution, which an interpreter can understand and execute together with a processor
- Responsibilities
    - Assigning responsibilities of tasks to actors
    - Detailing information about steps
- Charting
    - Construct a process flowchart
- Exceptions
    - Exception handling
- Tests

)

-o-

Format for a **Graph:**

(

Nodes (is a set),

Edges (is a set).

)

-o-

Format for a **Diagram:**

(

o is a process node,

-> is an edge,

|is a fork or a join node,

D is a decision node,

M is a merge node.

)

-o-

Format for a **System:**

(

Input,

Component (is a set),

Output.

)

-o-

Format for a **Function:**

(

Name,

Domain and Range,

Assingment rule.

)

-o-

Format for **Electronic Data Interchange file:**

(

https://www.json.org/json-en.html

)

-o-

„

Format for a **Format:**

X:

(

Y

)

-o-„

Notes:

- X and Y are sequences of symbols