DESIGNING A NEW PROGRAMMING

LANGUAGE

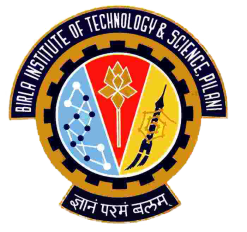
­­­

ASSIGNMENT

PRINCIPLES OF PROGRAMMING LANGUAGES (CS F301/IS F301)

UTKARSH PATHRABE 2012A7PS034P

SHIKHAR VASHISTH 2012C6PS436P

  
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

**DOMAIN OF PROGRAMMING LANGUAGE**

* For simplifying use, maintenance and automation of daily need equipments in houses, hostels, hospitals, traffic signals, prisons etc. as compared to designing with other available programming languages.
* Allows one to define all equipments along with their states and operations and to design hierarchical model of whole system. It has provision of automating certain processes. Also allows one to control all equipments individually or in groups through specifically designed command line instructions.

**PROBLEMS SOLVED BY THIS LANGUAGE**

Language can be used to computerize domains like-

* Managing various equipments of house like door, electronic equipments (fan, light sources, cleaner, television, air conditioners etc.) and allows them to work according to the sensors response or user commands. For example, one can program the fire alarm to ring automatically if heat sensor in kitchen detects high temperature. Similarly, allows to automatically switch off all electronic equipments if no one is present in the room.
* In hostels, language can be used to automatically switch OFF/ON lamp, fan in the room by detecting the presence of person in the room with the help of a sensor. It can also be used to program solar water heater to automatically switch ON when the sensor detects the temperature falling below a certain specified temperature.
* In prison, language can be used to automate alarm signals if prison break is detected. Sensors can be installed in every prison compartment to detect the presence of prisoners during specified time.
* The language can be used to manage general equipments in hospital like fan, lamp, TV etc. by the patient to put them ON or OFF as per his/her wish without the need to move around by enabling commands through sound input.
* For managing traffic on crossing. One can program sensors on road to detect over speeding vehicles and taking required actions. On crossings wait time can be changed according to traffic density at a given time.
* In factories, one can program various machines for doing their specific tasks and raising alarm when flaw in process is detected.

The language allows to write code performing such complex operations with less lines of code compared to the other programming languages available for these domains.

**PROGRAMMING FEATURES OF THIS LANGUAGE**

* **Declarative**
  + Allows to directly instruct the language what needs to be done.
* **Procedural**
  + Statements inside ‘AUTOMATE’ (periodic function) block are executed in a sequential manner.
* **Object Oriented**
  + Language allows to define every component as object that have data fields and associated methods.
* **Event driven**
  + Flow of programs is determined by events defined by user (e.g. Sensor output)
* **Structured**
  + Follows structured programming principles such as use of block structures and for loop in contrast to using simple tests and jumps.
* **Readable**
  + Syntax of language is such that it can be easily understood just by reading.
* **Writable**
  + Syntax of language is very similar to English language.
* **Abstraction**
  + Language allows to define complex processes and structures in an abstract way.
* **Support for parallelism**
  + Commands and automated processes defined through the language can be executed in parallel to improve performance and the support for this would be inherently provided by the interpreter.

**TOKENS USED IN LANGUAGE**

**Keywords:**

ABSTRACT\_TYPE It defines type of objects which can contain other concrete type objects. They themselves don’t have states and operations

CLASS It defines common states and operations which are inherited by concrete object types.

TYPE It defines concrete object type.

AUTOMATE To define processes which needs to get re-executed after certain time.

COMMAND To issue command to devices

DISPLAY Used to display message on console

**Data types:**

REAL Equivalent to double data type in C

INTEGER Equivalent to integer data type in C

LIST For storing possible values of type’s state

STRING For storing strings like in Java

**Special Operators:**

-> Equivalent to dot (.) operator in C

<- Assignment operator

& Refers to the parent of the object

@ Sets refresh rate for the automated process

\_ Used to define parameters of the functions   
like func \_ , \_ takes two parameters.

: Used to specify start of indentation

<TYPE> List all objects of specified TYPE in domain

|CLASS| List all objects of specified CLASS in domain

BETWEEN Takes two arguments after it, which are super types of the list of arguments before it and gives us the objects from the list which are common in both the parameters after it.

EXCEPT Takes one argument after it which needs to be excluded from the list of arguments before it.

IN For specifying domain in command

( ) Used to declare an array of concrete type containing specified objects

Like LAMP: bulb(5) declares an array of type LAMP of size 5.

# For single line comment

**Conditional constructs:**

IF condition THEN

Statements

ELSEIF condition THEN

Statements

ELSE

Statements

ENDIF

**Looping constructs:**

FOR variable IN list DO

Statements

ENDFOR

**Comparison Operator:**

< Less than

> Greater than

<= less than equal to

>= greater than equal to

= equal

!= not equal

**Arithmetic Operators:**

+ Addition

- Subtraction

\* Multiplication

/ Division

% Modulo

^ Exponentiation

**Logical Operators:**

AND Evaluates to true if both conditions are true

OR Evaluates to true if any one or both conditions are true

NOT Negates the condition

* Attribute names of concrete type should begin with an underscore (\_)
* AUTOMATE syntax:

**AUTOMATE** process\_name: domain(s)(separated by commas)  **@**Refresh\_rate(in ms)

Statements

**ENDAUTOMATE**

**SCENARIO-1: HOUSE**

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ABSTRACT TYPE INITIALIZATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*# Abstract Data Types for holding concrete objects*

**ABSTRACT\_TYPE** HOUSE

**ABSTRACT\_TYPE** FLOOR

**ABSTRACT\_TYPE** ROOM

**ABSTRACT\_TYPE** GARDEN

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CLASS INTIALIZATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*#Defining Classes which Typess can inherit*

**CLASS** ELECTRONIC: *#Defining Electronic class*

\_State LIST[ON, OFF] *#Stores the state of electronic device either ON or OFF*

SwitchOn *#Switches on the electronic device*

SwitchOff *#Switches off the electronic device*

**CLASS** MOBILE: *#Defining mobile class*

\_Position ABSTRACT\_TYPE *#Stores the information about position*

Goto \_ *#Orders object of mobile class to go to given location*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TYPE INITIALIZATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*#Defining concrete objects*

**TYPE** FAN: ELECTRONIC *#Defining type fan of electronic class*

\_Speed INTEGER *#Stores current speed of fan*

SpeedUp *#Increases the speed of fan by 1*

SpeedDown *#Decreases the speed of fan by 1*

**TYPE** LAMP: ELECTRONIC *#Defining type lamp of electronic class*

**TYPE** DOOR: *#Defining type door class*

\_State LIST[OPEN, CLOSE] *#Stores the state of door either OPEN or CLOSE*

Open *#Opens the door*

Close *#Closes the door*

**TYPE** ALARM: ELECTRONIC *#Defining type alarm of electronic class*

RingOn *#Rings the alarm*

RingOff *#Stops the ringing of alarm*

**TYPE** CLEANER: ELECTRONIC, MOBILE *#Defining type cleaner of electronic and mobile type*

StartCleaning *#Order cleaner to start cleaning*

**TYPE** TV: ELECTRONIC *#Defining type television of electronic class*

\_ChannelNumber INTEGER *#Stores current channel of tv*

\_VolumeLevel INTEGER *#Stores current volume level of tv*

ChannelUp *#Increases the channel number by 1*

ChannelDown *#Decreases the channel number by 1*

VolumeUp *#Increases the volume level by 1*

VolumeDown *#Decreases the volume level by 1*

**TYPE** FIRE\_DETECTOR: ELECTRONIC *#Defining type fire sensor of electronic class*

\_Temperature REAL *#Stores the temperature detected by sensor*

GetTemperature *#Gets the temperature from surroundings*

**TYPE** PRESENCE\_DETECTOR: ELECTRONIC *#Defining type presence detector of electronic type*

\_Presence LIST[HUMAN\_PRESENT, NO\_HUMAN, OWNER] *#Stores information about surroundings*

GetPresence *#Gets information about surroundings*

**TYPE** LIGHT\_DETECTOR: ELECTRONIC *#Defining type light detector of electronic type*

\_LightIntensity REAL *#Stores the light intensity detected*

GetLightIntensity *#Gets the light intensity from surroundings*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEFINING MAIN ARCHITECTURE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HOUSE**: MyHouse *#Defining the Architecture of House*

**DOOR**: Main\_gate *#House has one main gate*

**PRESENCE\_DETECTOR**: Owner\_detect *#HOUSE has a sensor which can detect presence*

**CLEANER**: Cleaner1 *#HOUSE has one cleaner named as Cleaner1*

**FLOOR**: Top\_floor *#House has a Top floor*

**DOOR**: Door1 *#Top floor has an entry door*

**ROOM**: Hall *#Top floor has one hall*

**DOOR**: Door1, Door2 *#Hall’s door1 links it to Top floor and door2 to kitchen*

**FAN**: Fan1, Fan2 *#Hall has two fans*

**LAMP**: Tubelight,Bulb *#Hall has tube light and bulb*

**TV**: LG\_tv *#Hall has a television*

**PRESENCE\_DETECTOR**: Human\_detect *#Hall has a presence sensor*

**ROOM**: Kitchen *#Top floor has a Kitchen*

**DOOR**: Door2 *#Kitchen has a door2 which links it to Hall*

**FAN**: Fan1 *#Kitchen has one fan*

**LAMP**: Cfl *#Kitchen has one CFL light*

**FIRE\_DETECTOR**: Fire\_detect *#Kitchen has a fire sensor*

**PRESENCE\_DETECTOR**: Human\_detect *#Kitchen has a presence sensor*

**ALARM**: Alm1 *#Kitchen has an alarm*

**FLOOR**: Ground\_floor *#House has a ground floor*

**DOOR**: Door3 *#Ground floor has entry door*

**ROOM**: Dining\_room *#Ground floor has dining room*

**DOOR**: Door3,Door4 *#Dining room’s door3 links to ground floor and door4 to common room*

**FAN**: Fan1 *#Dining room has a fan*

**LAMP**: Tubelight *#Dining room has a tube light*

**PRESENCE\_DETECTOR**: Human\_detect *#Dining room has a presence sensor*

**ROOM**: Common\_room *#Ground floor has common room*

**DOOR**: Door4 *#Common room’s door4 links it to Dining room*

**FAN**: Fan1 *#Common room has a fan*

**LAMP**: Bulb *#Common room has a bulb*

**PRESENCE\_DETECTOR**: Human\_detect *#Common room has a presence sensor*

**GARDEN**: Front\_garden *#House has a garden*

**LIGHT\_DETECTOR**: Daytime\_detect *#Garden has a light sensor*

**LAMP**: Bulb(4) *#Garden has 4 bulbs*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AUTOMATING PROCESSES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

*#Programs the fire sensor installed in kitchen to get temperature (in degree Fahrenheit) from surroundings after every 500ms and raise alarm if temperature is greater than 70 degree Celsius.*

**AUTOMATE** FIRE\_DETECTION: Kitchen **@** 500

Fire\_detect GetTemperature

REAL X <- Fire\_detect->\_Temperature

X <- (X – 32)\*5/9

**IF** X > 70 **THEN**

Alm1 RingOn

**ELSE**

Alm1 RingOff

**ENDIF**

**ENDAUTOMATE**

*#Programs the presence sensor of all rooms of top floor and ground floor to get information about surroundings after every 1sec and switch off all electronic devices if no human is present in that room*

**AUTOMATE** PRESENCE\_DETECTION: Top\_floor,Ground\_floor **@** 1000

**<**PRESENCE\_DETECTOR**>** GetPresence

**IF** PRESENCE\_DETECTOR \_Presence **=** NO\_HUMAN **THEN**

**|**ELECTRONIC**|** SwitchOff

**ENDIF**

**ENDAUTOMATE**

*#Programs the presence sensor installed outside the house to get information about surroundings after every 1sec and opens the main gate if detects the presence of owner.*

**AUTOMATE** ENTRY\_DETECTION: HOUSE **@** 1000

Owner\_detect GetPresence

**IF** Owner\_detect \_Presence **=** OWNER **THEN**

Main\_gate Open

**ENDIF**

**ENDAUTOMATE**

*#Programs the light sensor to get light intensity of surrounding after every half hour and turn off garden lights if its day else switch them ON if its night*

**AUTOMATE** GARDEN\_LIGHTS: Front\_garden @ 1800000

Daytime\_detect GetLightIntensity

**IF** Daytime\_detect \_LightIntensity < 100 **THEN**

<LAMP> SwitchOn

**ELSE**

<LAMP> SwitchOff

**ENDIF**

**ENDAUTOMATE**

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COMMAND LINE INSTRUCTIONS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*#Opens the Main\_gate*

**COMMAND**: Open Main\_gate

*#Opens all doors of Ground floor*

**COMMAND**: Open <DOOR> IN Ground\_floor

*#Opens all doors except Main\_gate*

**COMMAND**: Open <DOOR> EXCEPT Main\_gate

*#Among all doors it will open door between Hall and Kitchen*

**COMMAND**: Open <DOOR> BETWEEN Hall,Kitchen

*#Commands cleaner1 to turn on, go to the hall and start cleaning*

**COMMAND**: Cleaner1 SwitchOn AND Goto Hall AND StartCleaning

*#Switches off all electronic equipments in the house*

**COMMAND**: |ELECTRONIC| SwitchOff

*#Opens all doors of the house except the main gate*

**COMMAND**:

**FOR** X **IN** MyHouse**->** <DOOR> **DO**

**IF** X = MyHouse->Main\_gate **THEN**

X Close

**ELSE**

X Open

**ENDIF**

**ENDFOR**

**SCENARIO-2: HOSTEL**

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ABSTRACT TYPE INITIALIZATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*# Abstract Data Types for holding concrete objects*

**ABSTRACT\_TYPE** HOSTEL

**ABSTRACT\_TYPE** FLOOR

**ABSTRACT\_TYPE** ROOM

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CLASS INITIALIZATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*#Defining Classes which Typess can inherit*

**CLASS** ELECTRONIC: *#Defining Electronic class*

\_State LIST[ON, OFF] *#Stores the state of electronic device either ON or OFF*

SwitchOn *#Switches on the electronic device*

SwitchOff *#Switches off the electronic device*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TYPE INITIALIZATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*#Defining concrete objects*

**TYPE** FAN: ELECTRONIC *#Defining type fan of electronic class*

\_Speed INTEGER *#Stores current speed of fan*

SpeedUp *#Increases the speed of fan by 1*

SpeedDown *#Decreases the speed of fan by 1*

**TYPE** LAMP: ELECTRONIC *#Defining type lamp of electronic class*

**TYPE** DOOR: *#Defining type door class*

\_State LIST[OPEN, CLOSE] *#Stores the state of door either OPEN or CLOSE*

Open *#Opens the door*

Close *#Closes the door*

**TYPE** TV: ELECTRONIC *#Defining type television of electronic class*

\_ChannelNumber INTEGER *#Stores current channel of tv*

\_VolumeLevel INTEGER *#Stores current volume level of tv*

ChannelUp *#Increases the channel number by 1*

ChannelDown *#Decreases the channel number by 1*

VolumeUp *#Increases the volume level by 1*

VolumeDown *#Decreases the volume level by 1*

**TYPE** COOLER: ELECTRONIC

\_FanSpeed INTEGER *#Stores current speed of fan*

\_WaterLevel REAL *#Stores the water level of water*

\_PumpState LIST [ON, OFF] *#Stores the state of pump*

SpeedUp *#Increases the volume level by 1*

SpeedDown *#Decreases the channel number by 1*

PumpOn *#Switches on the pump*

PumpOff *#Switches off the pump*

**TYPE** ELECTRIC\_LOAD\_SENSOR: ELECTRONIC

\_ElectricLoad REAL *#Stores the value of electric load*

GetElectricLoad *#Gets the value of current electric load*

**TYPE** PRESENCE\_SENSOR: ELECTRONIC *#Defining type presence detector of electronic type*

\_Presence LIST[HUMAN\_PRESENT, NO\_HUMAN, OWNER]*#Stores information about surroundings* GetPresence *#Gets information about surroundings*

**TYPE** LIGHT\_SENSOR: ELECTRONIC *#Defining type light detector of electronic type*

\_LightIntensity REAL *#Stores the light intensity detected*

GetLightIntensity *#Gets the light intensity from surroundings*

**TYPE** TEMP\_SENSOR: ELECTRONIC *#Defining type temperature detector of electronic type*

\_Temperature REAL *#Stores the temperature detected*

GetTemperature *#Gets the temperature of surroundings*

**TYPE** SOLAR\_HEATER: ELECTRONIC *#Defining solar heater of electronic type*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEFINING MAIN ARCHITECTURE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HOSTEL**: Gandhi\_bhawan

**DOOR**: Main\_gate

**LAMP**: OutsideLights

**LIGHT**\_**SENSOR**: Daytime\_detect

**TEMP**\_**SENSOR**: Temp\_detect

**SOLAR**\_**HEATER**: Solar\_heater

**FLOOR**: Ground\_floor

**ROOM**: Room101

DOOR: Door\_101

**FAN:** Fan

**LAMP**: Tubelight, Bulb

**PRESENCE**\_**SENSOR**: Presence\_101

**ELECTRIC\_LOAD\_SENSOR**: ELoad\_101

**ROOM**: Room102

**DOOR**: Door\_102

**FAN**: Fan

**LAMP**: Tubelight, Bulb

**PRESENCE**\_**SENSOR**: Presence\_102

**ELECTRIC\_LOAD\_SENSOR**: ELoad\_102

**ROOM**: Room103

**DOOR**: Door\_103

**FAN**: Fan

**LAMP**: Tubelight, Bulb

**PRESENCE\_SENSOR**: Presence\_103

**ELECTRIC\_LOAD\_SENSOR**: ELoad\_103

**FLOOR**: Top\_floor

**ROOM**: Common\_room

**DOOR**: Door\_common1, Door\_common2

**FAN**: Fan1, Fan2, Fan3

**COOLER**: Cooler

**TV**: LG\_Tv

**LAMP**: Tube1,Tube2,Bulb

**PRESENCE**\_**SENSOR**: Presence\_common

**ROOM**: Room201

**DOOR**: Door\_201

**FAN**: Fan

**LAMP**: Tubelight, Bulb

**PRESENCE**\_**SENSOR**: Presence\_201

**ELECTRIC**\_**LOAD**\_**SENSOR**: ELoad\_201

**ROOM**: Room202

**DOOR**: Door\_202

**FAN**: Fan

**LAMP**: Tubelight, Bulb

**PRESENCE**\_**SENSOR**: Presence\_202

**ELECTRIC**\_**LOAD**\_**SENSOR**: ELoad\_202

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AUTOMATING PROCESSES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

*#Programs the presence sensor of all rooms of top floor and ground floor to get information about surroundings after every 1sec and switch off all electronic devices if no human is present in that room*

**AUTOMATE** PRESENCE\_DETECTION: Top\_floor,Ground\_floor **@** 1000

<PRESENCE\_SENSOR> GetPresence

**IF** PRESENCE\_DETECTOR \_Presence = NO\_HUMAN **THEN**

|ELECTRONIC| SwitchOff

**ENDIF**

**ENDAUTOMATE**

*#Programs the solar water heater to turn on and off automatically depending upon the temperature detected by the temperature sensor.*

**AUTOMATE** SOLAR\_CONTROL: Gandhi\_bhawan **@** 500

Temp\_detect GetTemperature

**REAL** X <- Temp\_detect->\_Temperature

**IF** X < 15 **THEN**

Solar\_heater SwitchOn

**ELSE** **IF** X > 30 **THEN**

Solar\_heater SwitchOff

**ENDIF**

**ENDAUTOMATE**

*#Programs the light sensor to get light intensity of surrounding after every half hour and turn off outside lights if its day else switch them ON if its night*

**AUTOMATE** OUTSIDE\_LIGHTS: Gandhi\_bhawan **@** 1800000

Daytime\_detect GetLightIntensity

**IF** Daytime\_detect \_LightIntensity < 100 **THEN**

OutsideLights SwitchOn

**ELSE**

OutsideLights SwitchOff

**ENDIF**

**ENDAUTOMATE**

*#Programs the electric load detecting sensors to automatically switch off all electronic devices of the room where load has exceeded certain limit and displays the name of room on the console of admin.*

**AUTOMATE** ELECTRIC\_LOAD\_DETECTION: Gandhi\_bhawan **@** 50000

**FOR** X **IN** <ELECTRIC\_LOAD\_SENSOR> **DO**

X GetElectricLoad

**IF** X->\_ElectricLoad > 1000 **THEN**

&X->|ELECTRONIC| SwitchOff

DISPLAY &X + " has exceeded Electric load limit."

**ENDIF**

**ENDFOR**

**ENDAUTOMATE**

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COMMAND LINE INSTRUCTIONS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*#Command closes all doors of entire bhawan and turns off all electronic devices*

**COMMAND**: Close <DOOR> **AND** SwitchOff |ELECTRONIC|

*#Command opens all doors and switches on all fans on top floor of bhawan.*

**COMMAND**: Open <DOOR> **AND** SwitchOn <FAN> **IN** Top\_floor

*#Command to open both doors of common room and to switch on cooler, TV, all lights and all fans of common room.*

**COMMAND**: Open Door\_common1, Door\_common2 **AND**

SwitchOn Cooler **AND**

SwitchOn LG\_Tv **AND**

SwitchOn <LAMP> **AND**

SwitchOn <FAN> **IN** Common\_room

*#Command to print the list of all rooms along with their current electric load.*

**COMMAND**: **FOR** X **IN** <ROOM> **DO**

X -> <ELECTRIC\_LOAD\_SENSOR> GetElectricLoad

**DISPLAY** X + “: ” + X -> <ELECTRIC\_LOAD\_SENSOR> -> \_ElectricLoad

**ENDFOR**