Lab Exercise 2- Understanding Model, Class, and Function in OMEdit (OpenModelica)

This lab introduces the core structural elements of Modelica:
model, **class**, and **function**.

PART A – Understanding MODEL in Modelica

Concept

A **model** in Modelica is used to describe dynamic systems with equations that can be simulated.

It contains:

- Parameters
- Variables
- Equations

A model is typically used for physical systems.

STEP 1: Create a Model

```
Open OMEdit → File → New Model
```

Name:

```
SimpleMassModel
```

Click OK.

STEP 2: Write a Basic Model

Add the following code:

```
model SimpleMassModel

  parameter Real m = 5;
  parameter Real g = -9.81;

  Real v(start = 0);
  Real h(start = 50);

equation

  der(v) = g;
  der(h) = v;

end SimpleMassModel;
```

STEP 3: Simulate the Model

Click:

- Check Model
- Simulate

Plot:

- h
- v

Observe free-fall motion.

PART B – Understanding CLASS in Modelica

Concept

In Modelica:

Everything is a class.

A model is a specialized type of class.

A **class** can define:

- Reusable components
- Data structures
- Parameter containers
- Base definitions

Classes may or may not be directly simulated.

STEP 4: Create a Class

```
File → New Model
```

Change "Restriction" to:

```
Class
```

Name:

```
VehicleParameters
```

Click OK.

STEP 5: Add Parameters in Class

```
class VehicleParameters



  parameter Real mass = 1000;

  parameter Real maxSpeed = 60;

  parameter Real enginePower = 150;



end VehicleParameters;
```

This class stores configuration data.

It is not directly simulated.

STEP 6: Use Class in a Model

Create a new model:

```
CarModel
```

Inside write:

```
model CarModel

  VehicleParameters vp;

  Real acceleration;


equation


  acceleration = vp.enginePower / vp.mass;


end CarModel;
```

Simulate and observe acceleration value.

Key Difference So Far

Model → Simulated dynamic system

Class → Reusable definition (data or structure)

PART C – Understanding FUNCTION in Modelica

Concept

A **function** performs a calculation and returns a value.

Functions:

- Do not contain differential equations
- Cannot contain der()
- Must use algorithm section
- Return outputs

Used for:

- Mathematical calculations
- Utility computations
- Supporting models

STEP 7: Create a Function

```
File → New Model
```

Change restriction to:

```
Function
```

Name:

```
KineticEnergy
```

Click OK.

STEP 8: Write Function Code

```
function KineticEnergy

  input Real m;
  input Real v;

  output Real KE;

algorithm

  KE := 0.5 * m * v^2;

end KineticEnergy;
```

STEP 9: Use Function in a Model

Create new model:

```
EnergyModel
```

Write:

```
model EnergyModel

  parameter Real m = 10;
  Real v(start = 5);
  Real KE;

equation

  der(v) = -9.81;
  KE = KineticEnergy(m, v);

end EnergyModel;
```

Simulate and plot:

- KE
- v

Observe kinetic energy change.

PART D – Comparison Table

Feature	Model	Class	Function
Used for	Dynamic systems	Reusable definitions	Calculations
Contains equations	Yes	Optional	No
Contains der()	Yes	Optional	No
Contains algorithm	Optional	Optional	Required
Simulatable	Yes	Only if model	No
Returns value	No	No	Yes