Some Notes on Object-Oriented Programming

Implementing a coordinate system in software

Data

A_{IC}
$_C\tilde{\omega}_{IC}$
color
3D_info
env
$\dot{A}_{IC}$

Functionality

$_I\vec{u} =$ transform (A_{IC}, $_C\vec{u}$)
$\dot{A}_{IC} =$ derivative (A_{IC}, $_C\tilde{\omega}_{IC}$)
3D_info = create3D_info (A_{IC})
showCoSys (color, 3D_info, env)

Object C combines data & functionality (state & behavior)

Create "Constructor" O = Classname (vars) → initialize
Delete "Destructor" O.delete() → clean-up

Members

→ can interact

	Data	Methods
Public	A_{IC} $_C\tilde{\omega}_{IC}$ color	constructor (A_{IC}, env, color) destructor() set(A_{IC}), set($_C\tilde{\omega}_{IC}$), ... transform($_C\vec{v}_C$) getDerivative()
Private	env 3D_info $\dot{A}_{IC}$	create3Dinfo() showCoSys() computeDerivative()

```
classdef MyCLASS < handle
   properties (SetAccess = public, GetAccess = public)
      myVariable  % a public variable
   end
   methods  % no arguments means 'public'
      function obj = MyCLASS(initialValue) % constructor
         obj.myVariable  = initialValue;   % initialize variable
      end
      function showVariable(obj)% 'obj' is always first variable
         disp(obj.myVariable)% access 'myVariable' of this object
      end
   end
end
```

```
% Construct object and initialize with 42:
obj1 = MyCLASS(42);
% Construct another object with 17
obj2 = MyCLASS(17);

obj1.showVariable(); % Output is 42
showVariable(obj1);  % (Alternative syntax)
obj2.showVariable(); % Output is 17

obj1.myVariable = 23;% Change state
obj1.showVariable(); % Output is 23
```