

CEE4408: Comparison between procedural programming and object oriented programming

% Procedural Programming

% AreaCalculation.m

```
function z = AreaCalculation (a, b)
```

```
z = a*b;
```

% VolumeCalculation.m

```
function z = VolumeCalculation (a, b, c)
```

```
z = a*b*c;
```

% CostCalculation.m

```
function z = CostCalculation (a, b, c, d)
```

```
z = a*b*c*d;
```

%main.m

```
ColumnName = ['Column A'; 'Column B'; 'Column C'];
```

```
Length = [1.1; 1.2; 1.3];
```

```
Width = [1.2; 1.3; 1.0];
```

```
Height = [5.2; 6.3; 7.0];
```

```
UnitCost = [20, 30, 25];
```

```
fi = fopen ('file1.dat','w');
```

```
% Plan area, volume, cost calculation
```

```
for n = 1:3
```

```
Area(n) = AreaCalculation(Length(n), Width(n));
```

```
Volume(n) = VolumeCalculation(Length(n), Width(n), Height(n));
```

```
Cost(n) = CostCalculation (Length(n), Width(n), Height(n), UnitCost(n));
```

```
fprintf (fi,'%10.8s %6.2f %6.2f %6.2f \n', ColumnName(n,:), Area(n), Volume(n),  
Cost(n));
```

```
end
```

```
fclose (fi);
```

% Object Oriented Programming

% Column.m

classdef Column

properties

Length;

Width;

Height;

UnitCost;

Area;

Volume;

Cost;

end

methods

function CO = Column(Length, Width, Height, UnitCost)

if nargin > 0

CO.Length = Length;

CO.Width = Width;

CO.Height = Height;

CO.UnitCost = UnitCost;

end

end

function Area = get.Area(CO)

Area = CO.Length * CO.Width;

end

function Volume = get.Volume(CO)

Volume = CO.Length * CO.Width * CO.Height;

end

function Cost = get.Cost(CO)

```
        Cost = CO.Length * CO.Width * CO.Height * CO.UnitCost;

    end

end

end

% Main2.m

ColumnA = Column(1.1, 1.2, 5.2, 20);

ColumnB = Column(1.2, 1.3, 6.3, 30);

ColumnC = Column(1.3, 1.0, 7.0, 25);


fprintf('ColumnA %6.2f %6.2f %6.2f \n', ColumnA.Area, ColumnA.Volume, ColumnA.Cost);
fprintf('ColumnB %6.2f %6.2f %6.2f \n', ColumnB.Area, ColumnB.Volume, ColumnB.Cost);
fprintf('ColumnC %6.2f %6.2f %6.2f \n', ColumnC.Area, ColumnC.Volume, ColumnC.Cost);
```