```
insert System ;
insert NewYork.Manhattan.House ;

Bill.House [Group Void pub] ==
{
	Start(Void) [Func Void pub static] ==
	{
		Bob [Kitchen.Assign(20 , 30)] ;
		BobNumPlates [Int] == Bob.GetNumPlates(Void) ;
		Console.Write('Bob's number of plates is ' + BobNumPlates) ;
		# if you use other constructor, you'll get an exception.
	}

	Kitchen(Void) [Class Void pri] ==
	{

		inherit Oven , Microwave ;

		Assign( NewNumPots [Int var] , NewNumPlates [Int con] ) [Func Void pub] ==
		{
			NumPots [Int var] == NewNumPots ;
			NumPlates [Int con] == NewNumPlates ;
		}

		Assign(Void) [Func Void pub]  ==
		{
			NumPots [Prim Int] == 10 ;
			StandardManhattanHouse [House] ;
			Bill'sFridge [Seq] == StandardManhattanHouse.GetFridge() ;
		}

		GetNumPlates(Void) [Func Int pub] ==
		{
			Console.NLWrite(NumPlates) ;
			output NumPlates ;
		}

	}

	Start [Class pub]
	{

		BobHeight [Int] == 180 ;

		if (BobHeight > 150 & BobHeight < 170) {
			Console.NLWrite('Average Height') ;
		}
		eif (x >|= 170 &|| Nationality ! 'alien' | Profile = 'tall') {
			Console.NLWrite('Tall') ;
		}
		eif (Void) {
			Console.NLWrite('Height Not Within Range') ;
		}

		lif (i [Int] == 0 , i += 1 , i <|= 4) {
			Console.NLWrite('Numbers 1 to 4, both inclusive') ;
		}

	}

}
```