class Device {

[string]$Brand

[string]$Model

[string]$VendorSku

[string]ToString(){

return ("{0}|{1}|{2}" -f $this.Brand, $this.Model, $this.VendorSku)

}

}

class Rack : Device {

hidden [int] $Slots = 8

static [Rack[]]$InstalledRacks = @()

[int]$Slots = 8

[string]$Brand

[string]$Model

[string]$VendorSku

[string]$AssetId

[Device[]]$Devices = [Device[]]::new($this.Slots)

Device(

[string]$b,

[string]$m,

[string]$vsk

){

$this.Brand = $b

$this.Model = $m

$this.VendorSku = $vsk

}

[void] AddDevice([Device]$dev, [int]$slot){

## Add argument validation logic here

$this.Devices[$slot] = $dev

}

[void]RemoveDevice([int]$slot){

## Add argument validation logic here

$this.Devices[$slot] = $null

}

[int[]] GetAvailableSlots(){

[int]$i = 0

return @($this.Devices.foreach{ if($\_ -eq $null){$i}; $i++})

}

}

$rack = [Rack]::new()

$surface = [Device]::new()

$surface.Brand = "Microsoft"

$surface.Model = "Surface Pro 4"

$surface.VendorSku = "5072641000"

$rack.AddDevice($surface, 2)

$rack

$rack.GetAvailableSlots()