

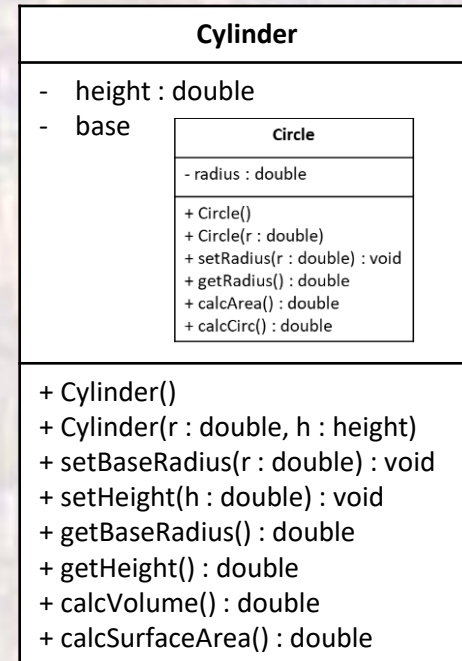
Composite Classes

Last updated 3/10/19

Composite Classes

- Composite Classes
 - Since objects are just like variables – objects can be member variables inside objects

```
class Cylinder{  
    // member data  
    private:  
        double height;  
        Circle base;  
  
    // member functions  
    public:  
        Cylinder();  
        Cylinder(double r, double h);  
        void setBaseRadius(double r);  
        void setHeight(double h);  
        double getBaseRadius(void);  
        double getHeight(void);  
        double calcVolume(void);  
        double calcSurfaceArea(void);  
}; // end Cylinder class
```



Composite Classes

- Composite Classes
 - Since objects are just like variables – objects can be member variables inside objects

```
#include "Cylinder.h"
```

```
#define PI 3.14159
```

```
Cylinder::Cylinder() {  
    base.setRadius(1);  
    height = 1;  
    return;  
}
```

```
Cylinder::Cylinder(double r, double h){  
    base.setRadius(r);  
    height = h;  
    return;  
}
```

```
void Cylinder::setBaseRadius(double r){  
    base.setRadius(r);  
    return;  
}
```

```
void Cylinder::setHeight(double h){  
    height = h;  
    return;  
}
```

```
double Cylinder::getBaseRadius(void){  
    return base.getRadius();  
}
```

```
double Cylinder::getHeight(void){  
    return height;  
}
```

```
double Cylinder::calcVolume(void){  
    return base.calcArea() * height;  
}
```

```
double Cylinder::calcSurfaceArea(void){  
    return (2*base.calcArea() + 2*PI*base.getRadius()*height);  
}
```

