Classes

Static Member Variables

Spring 2019

Instance Variables

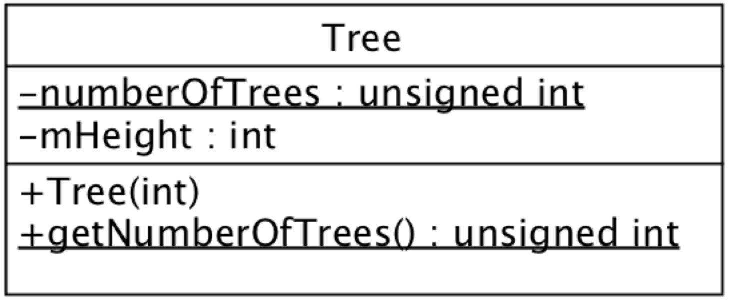
- Each object is an *instance* of a class
- Each object has its own copy of the member variables
- What does Rectangle cR1, cR2; look like in memory?

Static Members

- **static data members** and **static member functions** do not belong to any object

- Each object will access the same memory location

Static Member Example
Tree.h


Tree

–numberOfTrees : unsigned int
–mHeight : int

+Tree(int)
+getNumberOfTrees() : unsigned int

```
class Tree
{
  private:
     // static data member
    static unsigned int numberOfTrees;
    int mHeight;

  public:
    Tree (int height);

     // static member function
    static unsigned int getNumberOfTrees ();
};
```

Static Member Example
Tree.cpp

```
#include "Tree.h"

Tree::Tree (int height)
{
  mHeight = height;
  ++numberOfTrees;
}


// declaration and initialization

unsigned int Tree::numberOfTrees = 0;
```

Draw a picture of memory

```
Tree cOak(100);
Tree cMaple(97);
```

Static Member Functions

- Can only access **static** member variables
- Never marked const
- Call function with ::

```
static unsigned int getNumberOfTrees ();
```

Example

```
// Tree.cpp
unsigned int Tree::getNumberOfTrees ()
{
  return numberOfTrees;
}
```

```
// main.cpp
int main()
{
  Tree cOak;
  Tree cMaple;
  Tree cDouglasFir;

  cout << Tree::getNumberOfTrees() ; // ????
}
```

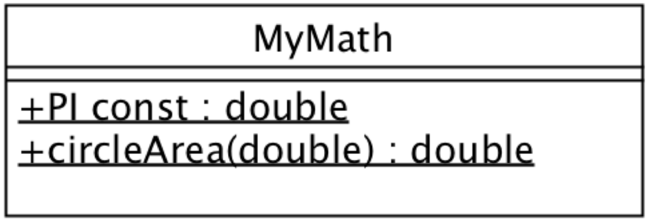
Problem

- Consider MyMath.h as follows:

```
#ifndef MYMATH_H
#define MYMATH_H

class MyMath
{
  public:
    static const double PI;
    static double circleArea (double);
};

#endif
```



MyMath

+PI const : double
+circleArea(double) : double

Problem

- Create MyMath.cpp as follows:

```
#include "MyMath.h"


const double MyMath::PI = 3.14159;


double MyMath::circleArea (double radius)
{
  // calculate the area of a circle


}
```

Problem

- Create MyMathDriver.cpp as follows:

```
#include <iostream>
#include "MyMath.h"


int main ()
{
  double radius;
  cout << "PI = " << MyMath::PI << endl;
  cout << "Enter the radius: ";
  cin >> radius;
  // Write the statement to output the area
  // of the circle
  return EXIT_SUCCESS;
}
```