C++: Static Members

Static Data Members

The idea: Having a single variable, regardless of how many instances of the class have been constructed.

Like a C-style global variable, but still part of a class. Benefits?

Declaring

Static variables are declared in the .h file along with the instance variables. Use the keyword static to distinguish them. Must be *redeclared* in the .cpp file.

Initializing

- constant static variables: in the .h file, with an initializer.
- non-constant static variables: in the .cpp file.
Accessing

From inside the class, access static variables like any other variable.

Two ways to accessing a static member from outside the class:

- Using an instance of the class:
  \[
  \text{max} = \text{td} \cdot \text{DESCR\_LENGTH};
  \]

- Using the class name:
  \[
  \text{max} = \text{TodoItem::DESCR\_LENGTH};
  \]

Which is better style?

Static Functions

Idea: Having a function that is not called on a particular instance of the class.

Like a C-style function, but still part of a class. Benefits?
Example

```cpp
// include "Account.h"
const double interest_rate = 4.0;
int Account::numAccounts = 0;
Account::Account()
{
    balance = BONUS;
}
int Account::numAccounts()
{
    return numAccounts;
}
double Account::deposit(int amount)
{
    balance += amount;
    return balance;
}
double Account::withdraw(int amount)
{
    balance -= amount;
    return balance;
}
static double changeBalance(double newRate)
{
    return balance * newRate / 100;
}
void Account::payInterest()
{
    balance = changeBalance(interest_rate);
}
int Account::getNumAccounts()
{
    return numAccounts;
}
```
// ----------------- Driver.cpp
#include <iostream.h>
#include "Account.h"

int main(void)
{

    Account a1;
    cout << "Starting balance: " << a1.getBalance() << "\n";
    a1.deposit(35);
    cout << "After deposit: " << a1.getBalance() << "\n";
    a1.payInterest();
    cout << "After interest: " << a1.getBalance() << "\n";

    Account::changeRate(50);
    a1.payInterest();
    cout << "After generous interest: "
         << a1.getBalance() << "\n";

    Account a2;
    a2.payInterest();
    cout << "Second account: " << a2.getBalance() << "\n";

    cout << "Number of accounts: "
         << Account::getNumAccounts() << "\n";
    cout<< "Bonus: " << Account::BONUS << "\n";

    return 0;
}