

```
// Demonstrate setprecision and constant
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
int main()  
{
```

```
    const double biweeklyDiv = 26.08857;
```

```
    float yearSal, biweekSal;
```

```
    cout << "What is your yearly salary? ";
```

```
    cin >> yearSal;
```

```
    biweekSal = yearSal / biweeklyDiv;
```

```
    cout << setprecision(2) << fixed;
```

```
    cout << "Your biweekly pay will be " <<
```

```
        biweekSal << endl;
```

```
    return 0;
```

```
}
```

```
// Demonstrate setprecision and constant
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
int main()  
{
```

```
    const double biweeklyDiv;
```

```
    biweeklyDiv = 26.08857;
```

```
    float yearSal, biweekSal;
```

```
    cout << "What is your yearly salary? ";
```

```
    cin >> yearSal;
```

```
    biweekSal = yearSal / biweeklyDiv;
```

```
    cout << setprecision(2) << fixed;
```

```
    cout << "Your biweekly pay will be " <<
```

```
        biweekSal << endl;
```

```
    return 0;
```

```
}
```

Why use `const` when a variable can do anything a `const` can do plus more?

`restrictions` can be your friend.

- confidence that no part of the program will change the value of the constant.

// Demonstrate use of left and right manipulators

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() {
```

```
    string month1 = "January", month2 = "February", ...;
```

```
    int days1 = 31, days2 = 28, days3 = 31, ...;
```

```
    double high1 = 6.9, high2 = 8.5;
```

```
    cout << fixed << showpoint << setprecision(1);
```

```
    cout << "Month      Days      High\n";
```

```
    cout << left << setw(12) << month1
```

```
        << right << setw(4) << days1
```

```
        << setw(9) << high1 << endl;
```

```
    ...
```

```
    return 0;
```

```
}
```