`TrackMomentumCalculator` interface changes

Kyle J. Knoepfel

LArSoft Coordination Meeting

6 November 2018

While debugging a segmentation violation…

I ran into the following in the `TrackMomentumCalculator` header file:

While debugging a segmentation violation…

I ran into the following in the `TrackMomentumCalculator` header file:

```
#include “iostream”

using namespace std;

// Global variables/input
Double_t xmeas[30]; Double_t ymeas[30]; Double_t eymeas[30]; Int_t nmeas;

class TrackMomentumCalculator {
// Problems:
//   - Many publicly accessible data members
//   - An extra 2 MB of unnecessary data members
//   - Unnecessary “setter” functions
};
```

While debugging a segmentation violation…

I ran into the following in the `TrackMomentumCalculator` header file:

```
#include "iostream"
```

→ *Will work, but non-standard; use* `<iostream>`

```
using namespace std;

// Global variables/input
Double_t xmeas[30]; Double_t ymeas[30]; Double_t eymeas[30]; Int_t nmeas;

class TrackMomentumCalculator {
// Problems:
//   - Many publicly accessible data members
//   - An extra 2 MB of unnecessary data members
//   - Unnecessary "setter" functions
};
```

While debugging a segmentation violation…

I ran into the following in the `TrackMomentumCalculator` header file:

```
#include "iostream"
```
→ *Will work, but non-standard; use `<iostream>`*

```
using namespace std;
```
→ ***Never** encouraged at (global) namespace scope.*

```
// Global variables/input
Double_t xmeas[30]; Double_t ymeas[30]; Double_t eymeas[30]; Int_t nmeas;

class TrackMomentumCalculator {
// Problems:
//   - Many publicly accessible data members
//   - An extra 2 MB of unnecessary data members
//   - Unnecessary "setter" functions
};
```

While debugging a segmentation violation…

I ran into the following in the `TrackMomentumCalculator` header file:

```
#include "iostream"
```
→ *Will work, but non-standard; use `<iostream>`*

```
using namespace std;
```
→ ***Never** encouraged at (global) namespace scope.*

```
// Global variables/input
Double_t xmeas[30]; Double_t ymeas[30]; Double_t eymeas[30]; Int_t nmeas;
```
→ ***Even worse. Do not define global variables in header files.***

```
class TrackMomentumCalculator {
// Problems:
//   - Many publicly accessible data members
//   - An extra 2 MB of unnecessary data members
//   - Unnecessary "setter" functions
};
```


Fermilab

While debugging a segmentation violation…

I ran into the following in the `TrackMomentumCalculator` header file:

```
#include "iostream"
```
→ *Will work, but non-standard; use `<iostream>`*

```
using namespace std;
```
→ ***Never** encouraged at (global) namespace scope.*

```
// Global variables/input
Double_t xmeas[30]; Double_t ymeas[30]; Double_t eymeas[30]; Int_t nmeas;
```
→ ***Even worse. Do not define global variables in header files.***

```
class TrackMomentumCalculator {
// Problems:
//   - Many publicly accessible data members
//   - An extra 2 MB of unnecessary data members
//   - Unnecessary "setter" functions
};
```

We sometimes make bad decisions, especially when we're beginners. However, some of these errors have existed in LArSoft for over 4 years.

Fermilab

Changes I have made (I propose LArSoft adopts these)

- Removed global variables and the “using” directive
- No longer any publicly accessible data members
  - `TGraph` or `TPolyLine3D` members—still exist but are private
- Made many implementation functions private
- The `SetMinLength` and `SetMaxLength` functions have been removed—these variables are now set in the `TrackMomentumCalculator` constructor.
- Implemented on the following feature branches:
  - **larreco**:`feature/knoepfel_TrackMomentumCalculator_cleanup`
  - **lariatsoft**:`feature/knoepfel_TrackMomentumCalculator_cleanup`
  - **ubana**:`feature/knoepfel_TrackMomentumCalculator_cleanup`
  - **dunetpc**:`feature/knoepfel_TrackMomentumCalculator_cleanup`