Example of Application

- Application to compute weighted average and error
  - Application must accept an arbitrary number of input data
  - Each data has a central value x and uncertainty
  - Compute weighted average of input data and uncertainty on the average
- Possible extensions
  - Provide different averaging methods
  - Uncertainties could be also asymmetric ($x^{+\sigma 1}_{-\sigma 2}$)
  - Consider also systematic errors
  - Compute correlation coefficient and take it into account when computing the average and its uncertainty
  - Use ROOT to make histogram of data points and plot a coloured band to indicate the average and its uncertainty overlaid on the histogram

Possible implementation

```
// wgtavg.cc
#include <vector>
#include <iostream>

#include “Datum.h"// basic data object
#include “InputService.h" // class dedicated to handle input of data
#include “Calculator.h” // implements various algorithms

using std::cout;
using std::endl;

int main() {

 std::vector<Datum> dati = InputService::readDataFromUser();

  Datum r1 = Calculator::weightedAverage(dati);
  cout << “weighted average: “ << r1 << endl;

  Datum r2 = Calculator::arithmeticAverage(dati);

  return 0;
}
```

Interface of Classes

```
#ifndef Calculator_h
#define Calculator_h

#include <vector>
#include "Datum.h"

class Calculator {
 public:
  Calculator();

  static Datum
    weightedAverage(const std::vector<Datum>& dati);
  static Datum
    arithmeticAverage(const std::vector<Datum>& dati);

};
#endif
```

```
#ifndef Datum_h
#define Datum_h
// Datum.h
#include <iostream>

class Datum {
  public:
    Datum();
    Datum(double x, double y);
    Datum(const Datum& datum);
    double value()  ;
    double error()  ;
    double significance();
  private:
    double value_;
    double error_;
};
#endi
```

```
#ifndef InputService_h
#define InputService_h
#include <vector>
#include "Datum.h"

class InputService {
 public:
   InputService();
   static std::vector<Datum> readDataFromUser();
 private:
};
#endif
```

You see the interface but don't know how the methods are implemented!

Application for Weighted Average

```
// wgtavg.cc
#include <vector>
#include <iostream>

#include "Datum" // basic data object
#include "InputService" // class dedicated to handle input of data
#include "Calculator" // impelments various algorithms

using std::cout;
using std::endl;

int main() {

  std::vector<Datum> dati = InputService::readDataFromUser();

  Datum r1 = Calculator::weightedAverage(dati);
  cout << "weighted average: " << r1 << endl;

  Datum r2 = Calculator::arithmeticAverage(dati);

  return 0;
}
```

```
$ g++ -c InputService.cc
$ g++ -c Datum.cc
$ g++ -c Calculator.cc
$ g++ -o wgtavg wgtavg.cpp InputService.o Datum.o Calculator.o
```

Questions

- ▻ What about reading a file of data?
  - – how to communicate the file name and where?
    - ○ in main or in InputService?
- ▻ Do you need any arguments for these functions?
- ▻ Who should compute correlation?
  - – should be stored?
    - ○ if yes, where?
  - – should the data become an attribute of some object?
    - ○ If yes, in which class?
- ▻ what about generating pseudo-data to test our algorithms?
  - – where would this generation happen?
  - – in the main() method or in some class?