Lab Session: Inheritance and Polymorphism

Shahram Rahatlou

SAPIENZA
UNIVERSITÀ DI ROMA

http://www.roma1.infn.it/people/rahatlou/programmazione++/

Corso di Programmazione++

Roma, 26 May 2009

Plan for Remaining Lab Sessions

- Develop a simple fitting tool using inheritance and polymorphism
  - Classes to handle different fitting models
    - Linear
    - Gaussian
    - Constant
    - 2^{nd} order polynomial
    - Exponential
  - Handle different types of input
    - std::vector<Datum>
    - histogram (if we have time)
  - Different minimization schemes
    - Least χ^2 method
    - Maximum Likelihood (if we have time)

Scope of the Sessions

- Exercise concepts of inheritance and polymorphism
- Design appropriate classes to for common and simple interface for users
- Application should use abstract interface classes instead of concrete and specific classes
- Possibility of extending your application beyond initial implementation
  - Adding new functions or fitting models should be transparent and easy
    - No rewriting but adding new classes

Blueprint of Application

- User provides data either from keyboard or (preferred) file
  - data includes vales and errors
- Consider using random numbers to generate a large sample of data to use as input
- Use simple std::vector to keep copy of data to fit
  - You need a clas to lhold measurements that is value and error
- Create a fitting model, for example a line at the beginning
- Create an object to handle minimization scheme
  - least squares at the beginning
- fit data to the model with

How Many and Which Classes?

- Think about classes need to implement
  - Fitter
  - Fit Function
  - Minimization scheme
- Use abstract or pure interface classes to enforce policy and interaction scheme
  - How do you want the user to use your objects
  - Which functions are needed?
- How to model Fitter object?
- How to model Minimization Scheme object?
  - What is it and what purpose it serves?

Before Writing Code Ask Yourself...

- How should your main() look like
  - Consider simple dummy classes before implementing functionalities
- Focus on defining the interface first
- Think about dependency between objects
  - Should a Function fit a set of data?
  - Should data fit to a model?
  - Does data knows about functions and/or fitter?
  - Should function know about kind of data?

Suggestion on How to Proceed

1. work mostly (perhaps only) on definition of classes that you need and what the product will look like

2. implement base classes, Linear Function and Least Squares minimization scheme

3. add support for fitting exponential and/or Gaussian