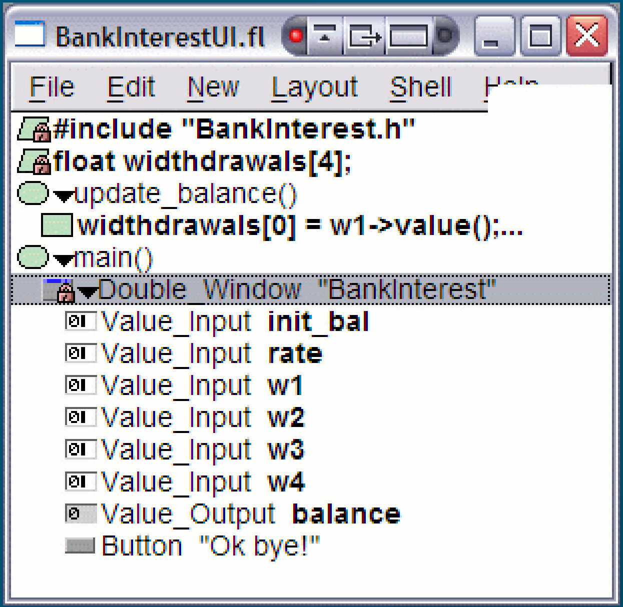
FLTK: Fast Light ToolKit

11 Feb 2009
CMPT166
Dr. Sean Ho
Trinity Western University

Compiling with GUI toolkits

- Libraries provide GUI components as objects
  - Windows (Fl_Window), menus, tabs, etc.
  - Widgets: buttons, textboxes (Fl_Input), sliders, scrollbars, dials, etc.
- Link your program with the toolkit library
  - Static linking: libfltk.a
    - Needed objects are bundled into the executable
  - Dynamic linking: libfltk.dll.a / libfltk.so
    - Need separate shared library
  - FLTK-1 libs: fltk, fltk_gl, fltk_images, fltk_forms

Using Fluid

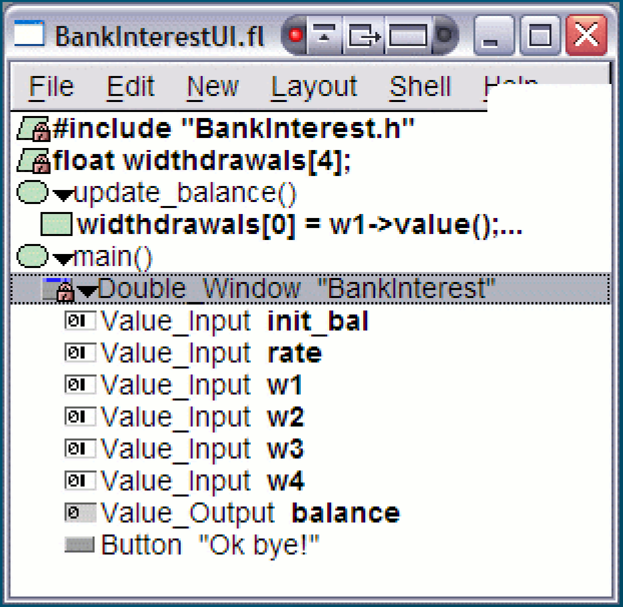


```
#include "BankInterest.h"
float widthdrawals[4];
update_balance()
  widthdrawals[0] = w1->value();...
main()
  Double_Window "BankInterest"
    Value_Input init_bal
    Value_Input rate
    Value_Input w1
    Value_Input w2
    Value_Input w3
    Value_Input w4
    Value_Output balance
    Button "Ok bye!"
```

- Fluid is FLTK's interactive GUI designer
  - Drag and drop widgets
  - Write code blocks / callbacks
- Saves *.fl Fluid files; exports *.cxx/*.h code
- Compile and link this code into your program
- It is possible to write a whole program in Fluid
- But better to separate GUI from main program logic: form vs. function
  - Akin to HTML/CSS vs. PHP/ASP/JS

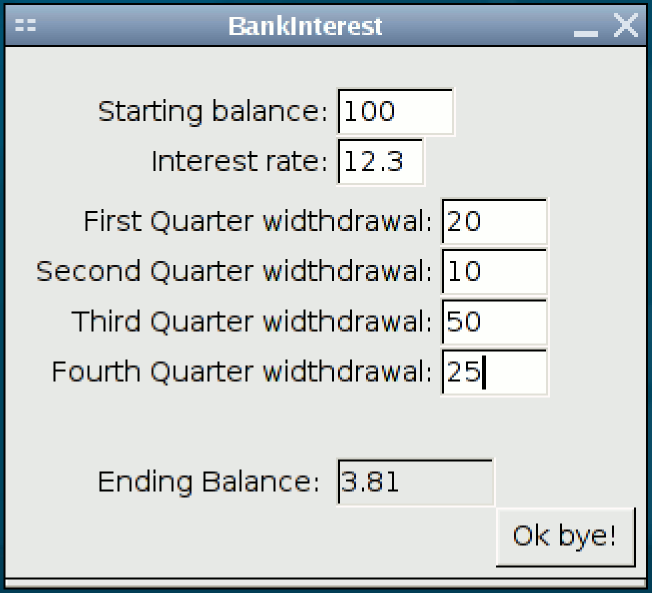
FLTK example: BankInterest

- BankInterestUI:
  - Just the user interface
  - Get values from the widgets
  - Minimal program logic
    - But I did choose to put main() here



```
#include "BankInterest.h"
float widthdrawals[4];
update_balance()
  widthdrawals[0] = w1->value();...
main()
  Double_Window "BankInterest"
    Value_Input init_bal
    Value_Input rate
    Value_Input w1
    Value_Input w2
    Value_Input w3
    Value_Input w4
    Value_Output balance
    Button "Ok bye!"
```

- BankInterest:
  - Main program functionality
  - Provides functions invoked by UI callbacks
    - calc_balance()



Starting balance: 100
Interest rate: 12.3
First Quarter widthdrawal: 20
Second Quarter widthdrawal: 10
Third Quarter widthdrawal: 50
Fourth Quarter widthdrawal: 25
Ending Balance: 3.81

UI design principles

- Constantine+Lockwood http://www.foruse.com/

- Structure: hierarchy, layout: windows, tabs, etc.
- Simplicity: make common tasks easy
  - Epicentre: design around primary purpose
- Visibility: need-to-know basis
- Feedback: current state, errors, etc.
- Tolerance: be flexible to user mistakes, save user data / user's hard work
- Reuse: consistent naming, behaviour