Lecture Qt003
Layouts

Instructor: David J. Coe

CPE 353 – Software Design and Engineering

Department of Electrical and Computer Engineering

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Outline

- Layouts
  - Horizontal
  - Vertical
  - Grid
- Exercise: Intro to Qt Creator IDE
- Key Points

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Layouts

- Layouts simplify development by taking responsibility for the position and size of its widgets
  - No need for hard-coded positions
  - Smooth, automatic resizing of windows

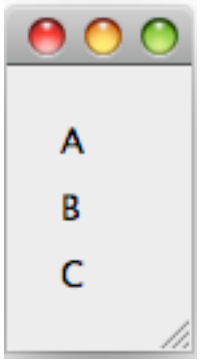
Horizontal Layout Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// hlayout.cpp -- Horizontal layout example
#include <QApplication>
#include <QWidget>
#include <QHBoxLayout>
#include <QLabel>
int main(int argc, char* argv[])
{
    QApplication    myApp(argc, argv);

    QWidget         widget;         // Primary display widget
    QHBoxLayout     mainlayout;     // Create horizontal layout object

    widget.setLayout(&mainlayout);  // Designate widget layout

    QLabel      label1("A");        // Create three labels
    QLabel      label2("B");
    QLabel      label3("C");

    mainlayout.addWidget(&label1);  // Make mainlayout responsible for
    mainlayout.addWidget(&label2);  // label appearance
    mainlayout.addWidget(&label3);

    widget.show();

    return  myApp.exec();
} // End main()
```

Vertical Layout Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

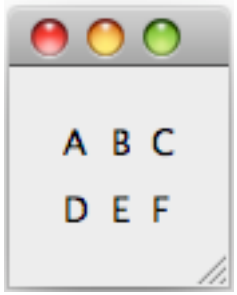



```
// vlayout.cpp -- Vertical layout example
#include <QApplication>
#include <QWidget>
#include <QVBoxLayout>
#include <QLabel>
int main(int argc, char* argv[])
{
    QApplication   myApp(argc, argv);

    QWidget        widget;          // Primary display widget
    QVBoxLayout    mainlayout;      // Create vertical layout object

    widget.setLayout(&mainlayout);  // Designate widget layout

    QLabel      label1("A");        // Create three labels
    QLabel      label2("B");
    QLabel      label3("C");

    mainlayout.addWidget(&label1);  // Make mainlayout responsible for
    mainlayout.addWidget(&label2);  // label appearance
    mainlayout.addWidget(&label3);

    widget.show();
    return  myApp.exec();

} // End main()
```

Grid Layout Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// glayout.cpp -- Grid layout example
#include <QApplication>
#include <QWidget>
#include <QGridLayout>
#include <QLabel>
int main(int argc, char* argv[])
{
    QApplication    myApp(argc, argv);

    QWidget         widget;              // Primary display widget
    QGridLayout     mainlayout;          // Create grid layout object
    widget.setLayout(&mainlayout);       // Designate widget layout

    QLabel      label1("A"), label2("B"); // Create six labels
    QLabel      label3("C"), label4("D");
    QLabel      label5("E"), label6("F");

    mainlayout.addWidget(&label1, 0, 0); // Make mainlayout responsible for
    mainlayout.addWidget(&label2, 0, 1); // label appearance
    mainlayout.addWidget(&label3, 0, 2);
    mainlayout.addWidget(&label4, 1, 0);
    mainlayout.addWidget(&label5, 1, 1);
    mainlayout.addWidget(&label6, 1, 2);

    widget.show();                       // Make widget visible

    return  myApp.exec();                // Start event loop
} // End main()
```



In-class Exercise:
Intro to Qt Creator IDE

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

- In a Linux terminal window, enter **qtcreator**
- Use Qt Creator to replicate Grid Layout Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Key Points

- Layouts responsible for presenting widgets in an orderly fashion
- Layouts may be manually created via the command line
- In most cases, you will want to use the **Designer** component of **Qt Creator** to quickly add widgets and apply layouts
- When using **Qt Creator** to create more complex and nested layouts, you may find it necessary to **break** and reapply layouts to make adjustments