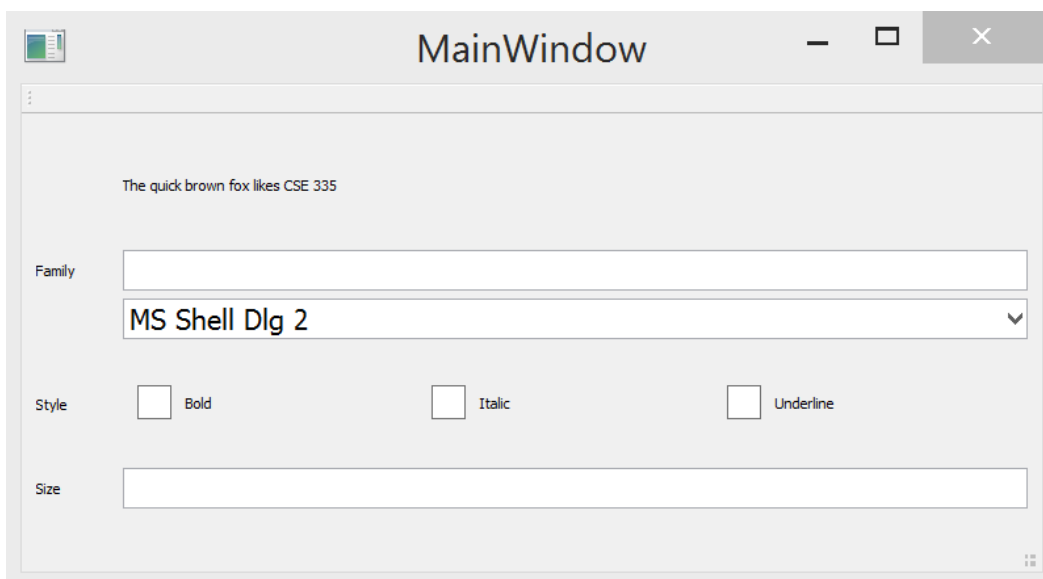


# Distributed Collaboration

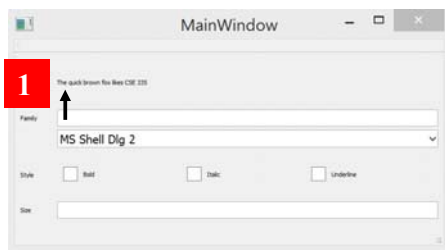
Alex X. Liu

**OO Software = Configuration + Collaboration**

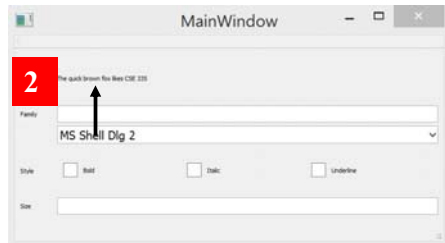
---



# Signals and Slots

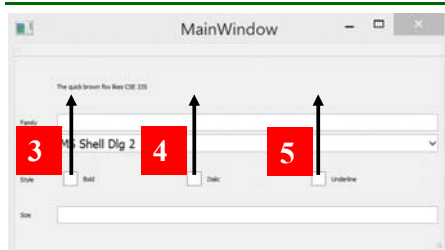


- QLineEdit signal: textChanged(QString&)
- QLabel slot: setText(QString&)
- Looks perfect! Are they the same QString?
  - NO!
  - Signal string: MS Shell Dlg 2
  - Slot string: The quick brown fox likes CSE 335.

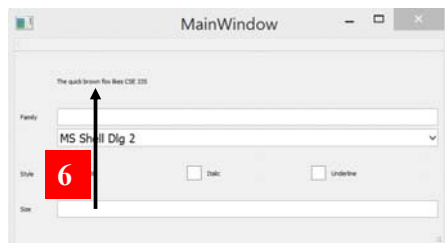


- QFontComboBox signal: currentFontChanged(QFont&)
- QLabel slot: setText(QString&)
- Even type does not match!

# Signals and Slots

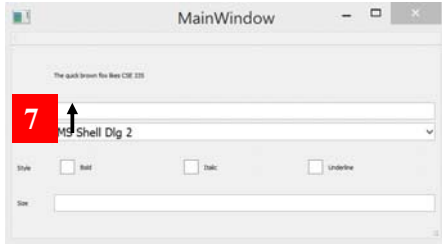


- QCheckBox signal: clicked(bool)
- QLabel slot: setText(QString&)
- Even type does not match!

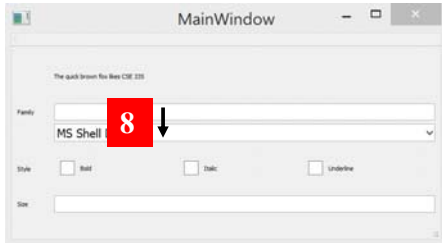


- QLineEdit signal: textChanged(QString&)
- QLabel slot: setText(QString&)
- Looks perfect! Are they the same QString?
  - NO!
  - Signal string: 32
  - Slot string: The quick brown fox likes CSE 335.

# Signals and Slots



- QFontComboBox signal: currentFontChanged(QFont&)
- QLineEdit slot: setText(QString&)
- Even type does not match!



- QLineEdit signal: textChanged(QString&)
- QFontComboBox slot: setCurrentFont(QFont&)
- Even type does not match!

## Solution: Design New GUI Classes with Needed Slots

- Problem: existing GUI class does not have the slot that you want.
- Solution:
  - 1. Make a new class that derive from the GUI class.
  - 2. Add the slots that you want in the new class.
- Another problem: in QT Designer, how to specify a GUI element is an object of the new class that you designed?
  - Right click the GUI object in the canvas.
  - Choose “Promote to” ->Your new class name.
  - In future, you can choose “Demote to” old class name.

# alexlabel.h

---

```
#ifndef ALEXLABEL_H
#define ALEXLABEL_H
#include <QObject>
#include <QLabel>
#include <QFont>
#include <QMessageBox>
/*Lessons:
 * 1. For any class with signals or slots, it must have the Q_OBJECT macro.
 * 2. Must separate .h and .cpp file. Otherwise you will get "undefined reference to vtable" error.*/
class AlexLabel: public QLabel{
    Q_OBJECT
public:
    AlexLabel(QWidget* qw):QLabel(qw){};
    //virtual ~AlexLabel(){};
public slots:
    //For receiving signal from QFontComboBox: void currentFontChanged(const QFont& font)
    void setFont(QFont);
    //For receiving signal QLineEdit for font family: void textChanged(const QString& text)
    void setFontByText(QString);
    //For receiving signal from CheckBox for Bold check box: void clicked(bool)
    void modifyFontByCheckboxBold(bool);
    //For receiving signal from CheckBox for Italic check box: void clicked(bool)
    void modifyFontByCheckboxItalic(bool);
    //For receiving signal from CheckBox for Underline check box: void clicked(bool)
    void modifyFontByCheckboxUnderline(bool);
    //For receiving signal QLineEdit for font size: void textChanged(const QString& text)
    void modifyFontByCheckboxSize(QString);
};
#endif // ALEXLABEL_H
```

---

Alex X. Liu

7

# alexlineedit.h

---

```
#ifndef ALEXLINEEDIT_H
#define ALEXLINEEDIT_H
#include <QLineEdit>

class AlexLineEdit: public QLineEdit{
    Q_OBJECT

public:
    AlexLineEdit(const QString& qstring):QLineEdit(qstring){};
    AlexLineEdit(QWidget* qw):QLineEdit(qw){};

public slots:
    //QFontCombo emit QFont signal, but QLineEdit slots only accept QString signal, so we have to make our own
    //slot to receive QFont signal, and then convert to QString.
    void showFont(QFont);
};

#endif // ALEXLINEEDIT_H
```

---

Alex X. Liu

8

# mainwindow.h

---

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QFont>
#include "alexlabel.h"

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

public slots:
    void getFont(QFont qf); // for the indirect signal-proxy-slot model

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H
```

---

Alex X. Liu

9

# mainwindow.cpp

---

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "alexlabel.h"
#include <QtGui>
#include <QMessageBox>
// #include <iostream>

// for the indirect signal-proxy-slot model
void MainWindow::getFont(QFont qf){
    ui->labelFox->setFont(qf);
    // QMessageBox::about(this, "test", qf.toString());
}

MainWindow::~MainWindow()
{
    delete ui;
}
```

---

Alex X. Liu

10

# mainwindow.cpp

---

```
MainWindow::MainWindow(QWidget *parent) :
```

```
    QMainWindow(parent),  
    ui(new Ui::MainWindow)
```

```
{  
    ui->setupUi(this);
```

```
1 connect(ui->lineEditFamily,SIGNAL(textChanged(QString)),ui->labelFox,SLOT(setFontByText(QString)));  
8 connect(ui->lineEditFamily,SIGNAL(textChanged(QString)),ui->fontComboBox,SLOT(setCurrentText(QString)));  
2 connect(ui->fontComboBox,SIGNAL(currentFontChanged(QFont)),ui->labelFox,SLOT(setFont(QFont)));  
  // for the indirect signal-proxy-slot model  
  //connect(ui->fontComboBox,SIGNAL(currentFontChanged(QFont)),this,SLOT(getFont(QFont)));  
7 connect(ui->fontComboBox,SIGNAL(currentFontChanged(QFont)),ui->lineEditFamily,SLOT(showFont(QFont)));  
3 connect(ui->checkBoxBold,SIGNAL(clicked(bool)),ui->labelFox,SLOT(modifyFontByCheckboxBold(bool)));  
4 connect(ui->checkBoxItalic,SIGNAL(clicked(bool)),ui->labelFox,SLOT(modifyFontByCheckboxItalic(bool)));  
5 connect(ui->checkBoxUnderline,SIGNAL(clicked(bool)),ui->labelFox,SLOT(modifyFontByCheckboxUnderline(bool)));  
6 connect(ui->lineEditSize,SIGNAL(textChanged(QString)),ui->labelFox,SLOT(modifyFontByCheckboxSize(QString)));  
}
```

---

Alex X. Liu

11

# alexlabel.cpp

---

```
#include "alexlabel.h"
```

```
#include <QFont>
```

```
void AlexLabel::setFont(QFont qf){
```

```
    QLabel::setFont(qf);
```

```
    //QMessageBox::about(this, "test", qf.toString());  
}
```

```
void AlexLabel::setFontByText(QString qs){
```

```
    this->setFont(QFont(qs));  
}
```

```
void AlexLabel::modifyFontByCheckboxBold(bool b){
```

```
    QFont font=this->font();
```

```
    font.setWeight(0); //Weight { Light, Normal, DemiBold, Bold, Black }
```

```
    font.setBold(b);
```

```
    this->setFont(font);
```

```
    //QMessageBox::about(this, "test", font.toString());  
}
```

```
void AlexLabel::modifyFontByCheckboxItalic(bool b){
```

```
    QFont font=this->font();
```

```
    font.setWeight(3); //Weight { Light, Normal, DemiBold, Bold, Black }
```

```
    font.setItalic(b);
```

```
    this->setFont(font);
```

```
    //QMessageBox::about(this, "test", font.toString());  
}
```

---

Alex X. Liu

12

# alexlabel.cpp

---

```
void AlexLabel::modifyFontByCheckboxUnderline(bool b){
    QFont font=this->font();
    font.setUnderline(b);
    this->setFont(font);
    //QMessageBox::about(this, "test", font.toString());
}

void AlexLabel::modifyFontByCheckboxSize(QString sizeString){
    //Get current font
    QFont font=this->font();
    //Modify the size of the font, keep other information of the font unchanged.
    int size=sizeString.toInt();
    if(size==0)
        font.setPointSize(12);
    else
        font.setPixelSize(size);
    //Use the modified font
    this->setFont(font);
}
```

# alexlineedit.cpp

---

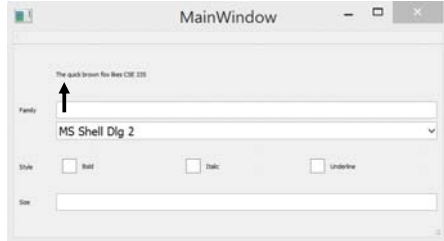
```
#include "alexlineedit.h"
```

```
void AlexLineEdit::showFont(QFont font){
    this->setText(font.family());
}
```

# Another Approach – By Proxy

---

- Two ways to solve the mismatch between signal and slot:
  - **Direct signal-slot model**: shown above.
  - **Indirect signal-proxy-slot model**: use mainwindow as the proxy!



- QLineEdit signal: textChanged(QString&)
- QLabel slot: setText(QString&)
- Proxy: mainwindow
  - Slot: LineEditFamilyChanged(QString&)
    - In this slot, 1. convert font name to new label string.
    - 2. Emit signal setLabelText(QString& newLabel)
  - Signal: setLabelText(QString& newLabel)
- In mainwindow constructor, two connects:
  - Connect QLineEdit signal textChanged(QString&) and mainwindow slot LineEditFamilyChanged(QString&)
  - Connect mainwindow setLabelText(QString& newLabel) signal and QLabel slot: setText(QString&)