13016209: Object-oriented Concepts and Programming

Second Semester, 2019

Project assignment - Guideline

---

1. Task

   A semester project implemented in C++ language

2. Objective

   The goal of this project assignment is to enhance students' skills and knowledge in object-oriented programming via C++ language. This is an individual assignment. Students can select their own interested project and develop it by using C++ language. Non-standard C++ classes or functions may be used in this project, but they must be identified and described. The project should be complicated enough so that the students can practice their analysis, design, and coding skills.

3. Submission

   Students are required to submit the followings:

   a. Project proposal – The proposal includes the student's name, project title, and project description. The project requirements should be clearly explained.

   b. Simple class diagram – Students must draw a simple class diagram that describes the classes, interfaces, and the relationship between classes (see Chapters 22 and 23 in Big C++ textbook)

   c. Completed program – Students need to demonstrate their program and submit the executable file of the program as well as the source code. The students must be able to explain their own source code.

   d. Final report – Final report includes the project proposal, revised class diagrams, source code, a list of non-standard C++ classes or functions (with descriptions) used in this project.

4. Grading

   The following topics are considered in grading of this project:

   a. Technical aspect –The problem to be solved should be complicated enough and the program should be efficiently implemented.

   b. Object-oriented concepts – Appropriate classes, objects, and techniques in C++ should be used to solve the problem. Key features of object-oriented

programming such as encapsulation, inheritance, or polymorphism should be exploited whenever they are useful.

c. Readability – Keep good program style! Students are required to use descriptive identifiers and add comments to help clarify the code. The indentation within the program should be consistent. Each class or function is clearly defined for solving a task.

d. Reliability – The program should validate the inputs before execution of the further process. Exception handling (see Chapter 17 in Big C++) technique should be used to handle exception cases. The output of the program must be correct and reliable.

5. Important dates

   a. Apr 8, 2020 (Wed) – Submission of the proposal (a PDF file)

   b. Apr 8, 2020 (Wed) – Submission of the class diagram (a PDF file)

   c. May 5, 2020 (Tue) – Submission of the executable program (files), source code (source files), and final report (a PDF file).

6. Notes

   a. Late submission (proposal, diagram, etc.) affects the score.

   b. Source code copied from others is not acceptable.

13016209: Object-oriented Concepts and Programming

Second Semester, 2019

Project assignment - Proposal

1. Project developer

Student ID	Name
62011277	Thawanrat Atthawiwatkul

2. Project title

Real Price (Shopping Calculator)

3. Project description and requirements

Project description:

Real Price is a shopping calculator with 6 difference calculators in the application.

- Normal Calculator
- Discount Calculator
- Currency Converter
- Loan Calculator
- Unit price Calculator
- Sales tax Calculator

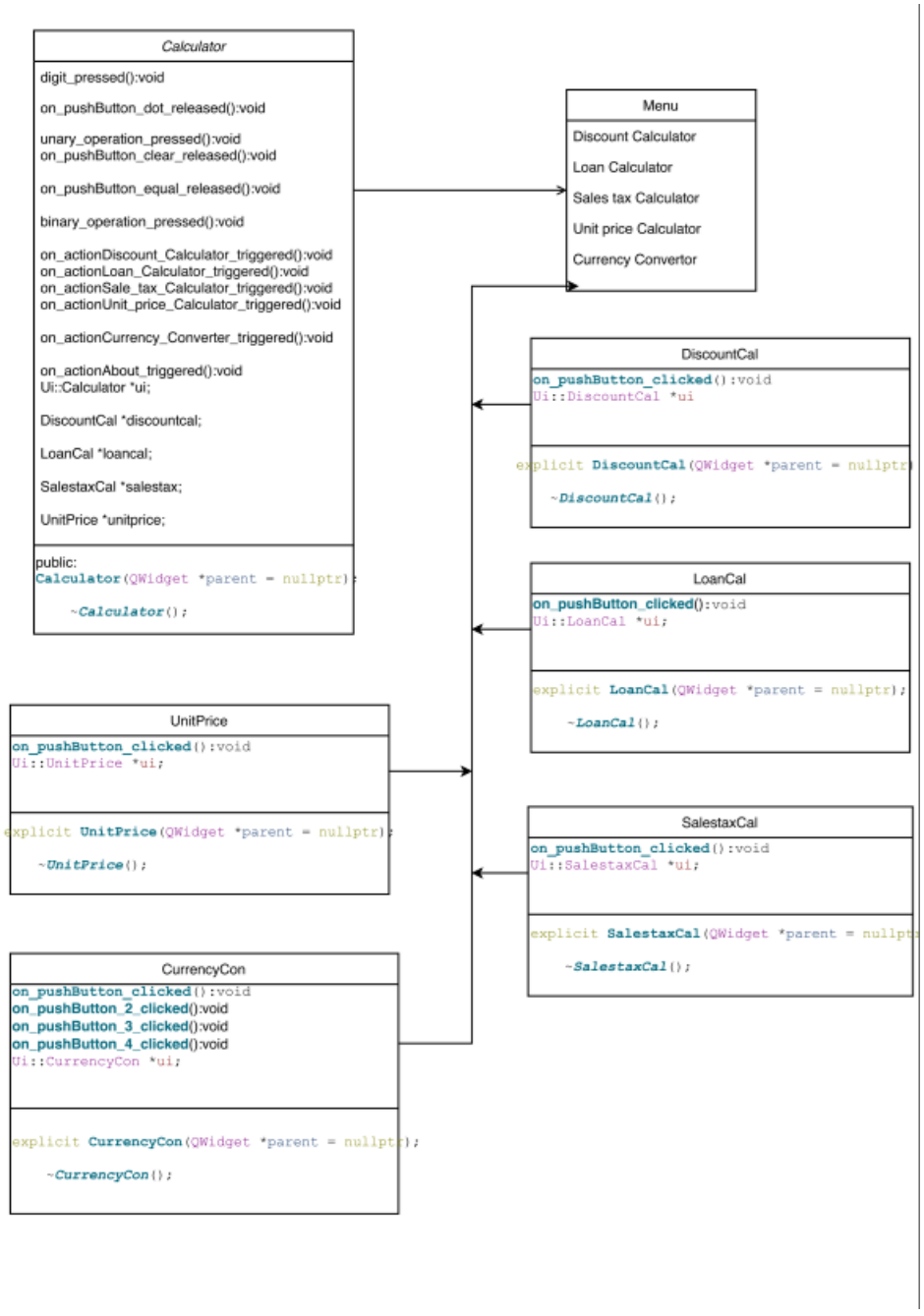
(Note: figures, flowcharts, diagrams may be used to describe the project)

Project requirements:

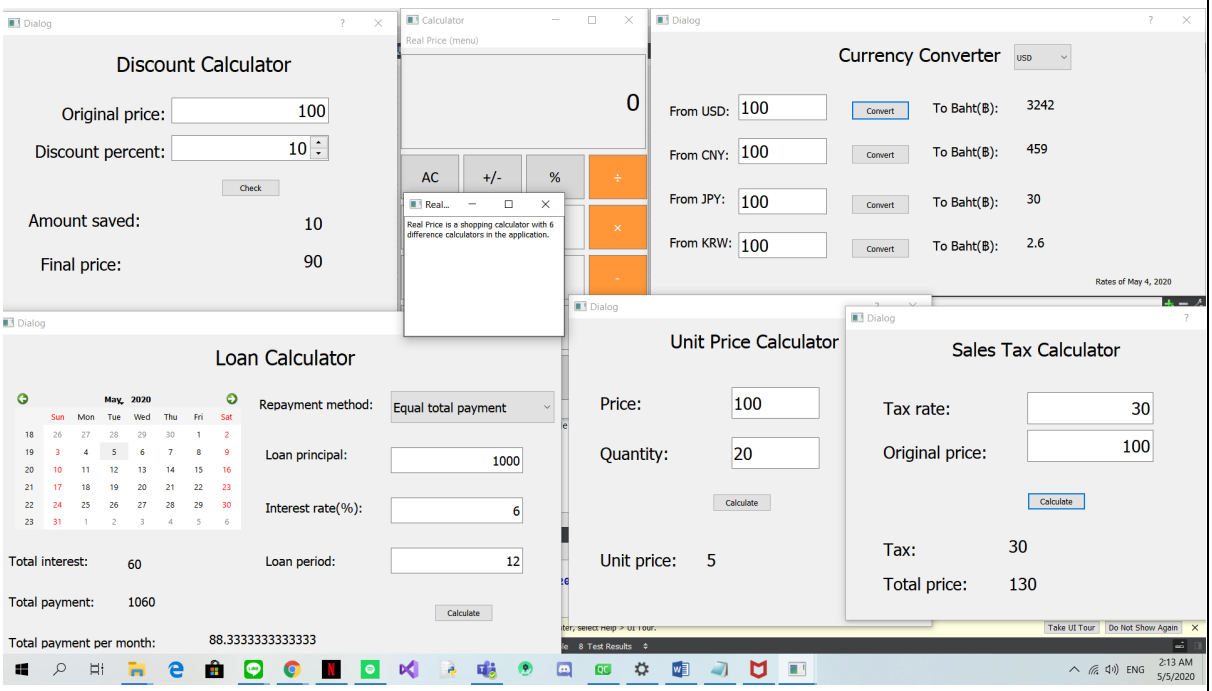
- User's input
- Qt Creator

(Describe the problem's statements or a specific request of the program that will be achieved.)

Class Diagram:



Output:



Source code:

In calculator.h :

```
#ifndef CALCULATOR_H
#define CALCULATOR_H

#include <QMainWindow>
#include <QTextEdit>
#include "discountcal.h"
#include "loancal.h"
#include "salestaxcal.h"
#include "unitprice.h"
#include "currencycon.h"

QT_BEGIN_NAMESPACE
namespace Ui { class Calculator; }
QT_END_NAMESPACE

class Calculator : public QMainWindow
{
    Q_OBJECT

public:
    Calculator(QWidget *parent = nullptr);
    ~Calculator();

private:
    Ui::Calculator *ui;
    DiscountCal *discountcal;
    LoanCal *loancal;
    SalestaxCal *salestax;
```

```
    UnitPrice *unitprice;
    CurrencyCon *currency;
private slots:
    void digit_pressed();
    void on_pushButton_dot_released();
    void unary_operation_pressed();
    void on_pushButton_clear_released();
    void on_pushButton_equal_released();
    void binary_operation_pressed();
    void on_actionDiscount_Calculator_triggered();
    void on_actionLoan_Calculator_triggered();
    void on_actionSale_tax_Calculator_triggered();
    void on_actionUnit_price_Calculator_triggered();
    void on_actionCurrency_Converter_triggered();
    void on_actionAbout_triggered();
};
#endif // CALCULATOR_H
```

In currencycon.h :

```
#ifndef CURRENCYCON_H
#define CURRENCYCON_H

#include <QDialog>

namespace Ui {
class CurrencyCon;
}

class CurrencyCon : public QDialog
{
    Q_OBJECT

public:
    explicit CurrencyCon(QWidget *parent = nullptr);
    ~CurrencyCon();

private slots:

    void on_pushButton_clicked();

    void on_pushButton_2_clicked();

    void on_pushButton_3_clicked();

    void on_pushButton_4_clicked();

private:
    Ui::CurrencyCon *ui;
};

#endif // CURRENCYCON_H
```

In discountcal.h :

```
#ifndef DISCOUNTCAL_H
#define DISCOUNTCAL_H

#include <QDialog>

namespace Ui {
class DiscountCal;
}

class DiscountCal : public QDialog
{
    Q_OBJECT

public:
    explicit DiscountCal(QWidget *parent = nullptr);
    ~DiscountCal();

private slots:
    void on_pushButton_clicked();

private:
    Ui::DiscountCal *ui;
};

#endif // DISCOUNTCAL_H
```

In loancal.h :

```
#ifndef LOANCAL_H
#define LOANCAL_H

#include <QDialog>

namespace Ui {
class LoanCal;
}

class LoanCal : public QDialog
{
    Q_OBJECT

public:
    explicit LoanCal(QWidget *parent = nullptr);
    ~LoanCal();

private slots:
    void on_pushButton_clicked();

private:
    Ui::LoanCal *ui;
};

#endif // LOANCAL_H
```

In salestaxcal.h :

```
#ifndef SALESTAXCAL_H
#define SALESTAXCAL_H

#include <QDialog>

namespace Ui {
class SalestaxCal;
}

class SalestaxCal : public QDialog
{
    Q_OBJECT

public:
    explicit SalestaxCal(QWidget *parent = nullptr);
    ~SalestaxCal();

private slots:
    void on_calculate_clicked();

private:
    Ui::SalestaxCal *ui;
};

#endif // SALESTAXCAL_H
```

In unitprice.h :

```
#ifndef UNITPRICE_H
#define UNITPRICE_H

#include <QDialog>

namespace Ui {
class UnitPrice;
}

class UnitPrice : public QDialog
{
    Q_OBJECT

public:
    explicit UnitPrice(QWidget *parent = nullptr);
    ~UnitPrice();

private slots:
    void on_pushButton_clicked();

private:
    Ui::UnitPrice *ui;
};

#endif // UNITPRICE_H
```

In calculator.cpp :

```
#include "calculator.h"
#include "ui_calculator.h"
#include "discountcal.h"
//#include <qmessagebox.h>

//#include <QtDebug>
double firstnum; //global variabal
bool usertypesecondnum = false;

Calculator::Calculator(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::Calculator)
{
    ui->setupUi(this);

    //for numbers 0-9
    connect(ui-
>pushButton_0,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_1,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_2,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_3,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_4,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_5,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_6,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_7,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_8,SIGNAL(released()),this,SLOT(digit_pressed()));
    connect(ui-
>pushButton_9,SIGNAL(released()),this,SLOT(digit_pressed()));

    //for operator %,+/-
    connect(ui-
>pushButton_plusminus,SIGNAL(released()),this,SLOT(unary_operation_pressed
()));
    connect(ui-
>pushButton_percent,SIGNAL(released()),this,SLOT(unary_operation_pressed()
));
    //for operator +,-,*,/
    connect(ui-
>pushButton_plus,SIGNAL(released()),this,SLOT(binary_operation_pressed()))
;
    connect(ui-
>pushButton_minus,SIGNAL(released()),this,SLOT(binary_operation_pressed())
);
    connect(ui-
>pushButton_multiply,SIGNAL(released()),this,SLOT(binary_operation_pressed
()));
    connect(ui-
>pushButton_divide,SIGNAL(released()),this,SLOT(binary_operation_pressed()
));
```

```
    ui->pushButton_plus->setCheckable(true);
    ui->pushButton_minus->setCheckable(true);
    ui->pushButton_multiply->setCheckable(true);
    ui->pushButton_divide->setCheckable(true);
}

Calculator::~Calculator()
{
    delete ui;
}

void Calculator::digit_pressed() {
    //qDebug() << "test";
    QPushButton *button = (QPushButton*)sender();
    double labelNumber;
    QString newlabel;

    if((ui->pushButton_plus->isChecked()||ui->pushButton_minus-
>isChecked()||ui->pushButton_multiply->isChecked()
            ||ui->pushButton_divide->isChecked()) && (!usertypesecondnum))
    {
        labelNumber = button->text().toDouble();
        usertypesecondnum = true;
        newlabel = QString::number(labelNumber,'g',15);
    }
    else
    {
        if(ui->label->text().contains('.') && button->text() == "0")
        {
            newlabel = ui->label->text() + button-> text();
        }
        else
        {
            labelNumber = (ui->label->text()+button->text()).toDouble();
            newlabel = QString::number(labelNumber,'g',15);
        }
        labelNumber = (ui->label->text() + button->text()).toDouble();
    }

    //labelNumber = (ui->label->text()+button->text()).toDouble();
    //newlabel = QString::number(labelNumber,'g',15);
    ui->label->setText(newlabel);

}

void Calculator::on_pushButton_dot_released()
{
    ui->label->setText(ui->label->text() + ".");
}

void Calculator::unary_operation_pressed()
{
    QPushButton * button = (QPushButton*) sender();
    double labelNumber;
    QString newlabel;

    if(button->text() == "+/-")
    {
        labelNumber = ui->label->text().toDouble();
        labelNumber = labelNumber * -1;
```

```
        newlabel = QString::number(labelNumber,'g',15);
        ui->label->setText(newlabel);
    }

    if(button->text() == "%")
    {
        labelNumber = ui->label->text().toDouble();
        labelNumber = labelNumber * 0.01;
        newlabel = QString::number(labelNumber,'g',15);
        ui->label->setText(newlabel);
    }

}

void Calculator::on_pushButton_clear_released()
{
    ui->pushButton_plus->setChecked(false);
    ui->pushButton_minus->setChecked(false);
    ui->pushButton_multiply->setChecked(false);
    ui->pushButton_divide->setChecked(false);

    usertypesecondnum = false;
    ui->label->setText("0");


}

void Calculator::on_pushButton_equal_released()
{
    double labelNumber, secondnum;
    QString newlabel;

    secondnum = ui->label->text().toDouble();

    if(ui->pushButton_plus->isChecked())
    {
       labelNumber = firstnum + secondnum;
       newlabel = QString::number(labelNumber,'g',15);
       ui->label->setText(newlabel);
       ui->pushButton_plus->setChecked(false);
    }
    else if(ui->pushButton_minus->isChecked())
    {
        labelNumber = firstnum - secondnum;
        newlabel = QString::number(labelNumber,'g',15);
        ui->label->setText(newlabel);
        ui->pushButton_minus->setChecked(false);

    }
    else if(ui->pushButton_multiply->isChecked())
    {
        labelNumber = firstnum * secondnum;
        newlabel = QString::number(labelNumber,'g',15);
        ui->label->setText(newlabel);
        ui->pushButton_multiply->setChecked(false);
    }
    else if(ui->pushButton_divide->isChecked())
    {
        labelNumber = firstnum / secondnum;
        newlabel = QString::number(labelNumber,'g',15);
```

```
        ui->label->setText(newlabel);
        ui->pushButton_divide->setChecked(false);
    }

    usertypesecondnum = false;
}

void Calculator::binary_operation_pressed()
{
    QPushButton * button = (QPushButton*) sender();
    firstnum = ui->label->text().toDouble();

    button->setChecked(true);
}

void Calculator::on_actionDiscount_Calculator_triggered()
{
    discountcal = new DiscountCal(this);
    discountcal->show();
}

void Calculator::on_actionLoan_Calculator_triggered()
{
    loancal = new LoanCal(this);
    loancal->show();
}

void Calculator::on_actionSale_tax_Calculator_triggered()
{
    salestax = new SalestaxCal(this);
    salestax->show();
}

void Calculator::on_actionUnit_price_Calculator_triggered()
{
    unitprice = new UnitPrice(this);
    unitprice->show();
}

void Calculator::on_actionCurrency_Converter_triggered()
{
    currency = new CurrencyCon(this);
    currency->show();
}

void Calculator::on_actionAbout_triggered()
{
    QTextEdit *message = new QTextEdit();
    message->setWindowFlag(Qt::Window);
    message->setReadOnly(true);
    message->append("Real Price is a shopping calculator with 6 difference
calculators in the application. by Thawanrat A. 62011277");
    message->show();

}
```

In currencycon.cpp :

```
#include "currencycon.h"
#include "ui_currencycon.h"

CurrencyCon::CurrencyCon(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::CurrencyCon)
{
    QString usd;
    ui->setupUi(this);
    //ui->comboBox->addItem("USD").toDouble();

}

CurrencyCon::~CurrencyCon()
{
    delete ui;
}

void CurrencyCon::on_pushButton_clicked()
{
    double usd,usdb;
    QString newlabel;

    usd = ui->usd->text().toDouble();
    usdb = usd * 32.42;

    newlabel = QString::number(usdb,'g',15);
    ui->usdb->setText(newlabel);
}

void CurrencyCon::on_pushButton_2_clicked()
{
    double cny,cnyb;
    QString newlabel;

    cny = ui->cny->text().toDouble();
    cnyb = cny * 4.59 ;

    newlabel = QString::number(cnyb,'g',15);
    ui->cnyb->setText(newlabel);
}

void CurrencyCon::on_pushButton_3_clicked()
{
    double jpy,jpyb;
    QString newlabel;

    jpy = ui->jpy->text().toDouble();
    jpyb = jpy * 0.30 ;

    newlabel = QString::number(jpyb,'g',15);
    ui->jpyb->setText(newlabel);
}

void CurrencyCon::on_pushButton_4_clicked()
{
    double krw,krwb;
    QString newlabel;
```

```
    krw = ui->krw->text().toDouble();
    krwb = krw * 0.026 ;

    newlabel = QString::number(krwb,'g',15);
    ui->krwb->setText(newlabel);

}
```

In discountcal.cpp :

```
#include "discountcal.h"
#include "ui_discountcal.h"

DiscountCal::DiscountCal(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::DiscountCal)
{
    ui->setupUi(this);
    //connect(ui-
>pushButton,SIGNAL(released()),this,SLOT(on_pushButton_clicked()));
}

DiscountCal::~DiscountCal()
{
    delete ui;
}

void DiscountCal::on_pushButton_clicked()
{
    double price,percent,saved,final;
    QString newlabel;
    QString newlable2;

    price = ui->oriprice->text().toDouble();
    percent = ui->dispercent->text().toDouble();
    percent = percent * 0.01;

    saved = price * percent;
    final = price * (1 - percent);

    newlabel = QString::number(saved,'g',15);
    ui->saved->setText(newlabel);
    newlabel = QString::number(final,'g',15);
    ui->finalprice->setText(newlabel);

}
```

In loancal.cpp :

```
#include "loancal.h"
#include "ui_loancal.h"

LoanCal::LoanCal(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::LoanCal)
{
    ui->setupUi(this);
    connect(ui-
>calculate,SIGNAL(released()),this,SLOT(on_pushButton_clicked()));
}

LoanCal::~LoanCal()
{
    delete ui;
}

void LoanCal::on_pushButton_clicked()
{
    double pay,rate,period,tinterest,tpay,tmonth;
    QString newlabel;
    QString newlable2;

    pay = ui->loanp->text().toDouble();
    period = ui->period->text().toDouble();
    rate = ui->rate->text().toDouble();
    rate = rate * 0.01;

    tinterest = pay * rate;
    tpay = pay * (1.00 + rate);
    tmonth = tpay / period;

    newlabel = QString::number(tinterest,'g',15);
    ui->tinterest->setText(newlabel);
    newlabel = QString::number(tpay,'g',15);
    ui->tpay->setText(newlabel);
    newlabel = QString::number(tmonth,'g',15);
    ui->tmonth->setText(newlabel);


}
```

In salestaxcal.cpp :

```
#include "salestaxcal.h"
#include "ui_salestaxcal.h"

SalestaxCal::SalestaxCal(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::SalestaxCal)
{
    ui->setupUi(this);
}

SalestaxCal::~SalestaxCal()
{
    delete ui;
```

```
}

void SalestaxCal::on_calculate_clicked()
{
    double taxrate,oriprice,tax,total;
    QString newlabel;

    oriprice = ui->oriprice->text().toDouble();
    taxrate = ui->taxrate->text().toDouble();
    taxrate = taxrate * 0.01;

    tax = oriprice* taxrate;
    total = oriprice * (1 + taxrate);

    newlabel = QString::number(tax,'g',15);
    ui->tax->setText(newlabel);
    newlabel = QString::number(total,'g',15);
    ui->total->setText(newlabel);
}
```

In unitprice.cpp :

```
#include "unitprice.h"
#include "ui_unitprice.h"

UnitPrice::UnitPrice(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::UnitPrice)
{
    ui->setupUi(this);
}

UnitPrice::~UnitPrice()
{
    delete ui;
}

void UnitPrice::on_pushButton_clicked()
{
    double price,quantity,unitprice;
    QString newlabel;

    price = ui->price->text().toDouble();
    quantity = ui->quantity->text().toDouble();

    unitprice = price / quantity;

    newlabel = QString::number(unitprice,'g',15);
    ui->unitprice->setText(newlabel);
}
```

In main.cpp :

```
#include "calculator.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    Calculator w;
    w.show();
    return a.exec();
}
```

List of non-standard classes :

- calculator.h
- currencycon.h
- discountcal.h
- loancal.h
- salestaxcal.h
- unitprice.h