

UEE1303(1009) S22: Homework 2

Due: 2022/6/20(Mon.) 23:55

[Instruction]

- Please put your source code files of each problem into separate folder named StudentID_hw2_1 and StudentID_hw2_2, compress these two folders to **zip** files separately (Ex: 110511000_hw2_1.zip, 110511000_hw2_2.zip), and upload these two zip files to e3 before deadline.
- **If zipped file's or source code file's name is wrong, your score of this homework is 30% off.**
- **Your source code files should be able to be compiled and executed on our server.**
- Your output should follow the example, otherwise you will not get full credit.
- If you have any question, please send an email to TA or leave a message in the line account.

[Problem 1]: RLC Series Impedance Calculator

In this problem, you should calculate the impedance of the series RLC circuits given then the sequence values of a resistor(Ω), an inductor(μF), a capacitor(mH) and a frequency(Hz) of a sinusoidal signal.

Example: A 1 mH inductor and a 100 μF capacitor, and a 1 Ω resistor at a frequency of 10 Hz. This example shows the near-resonance impedance of about 15.915430 k Ω s.

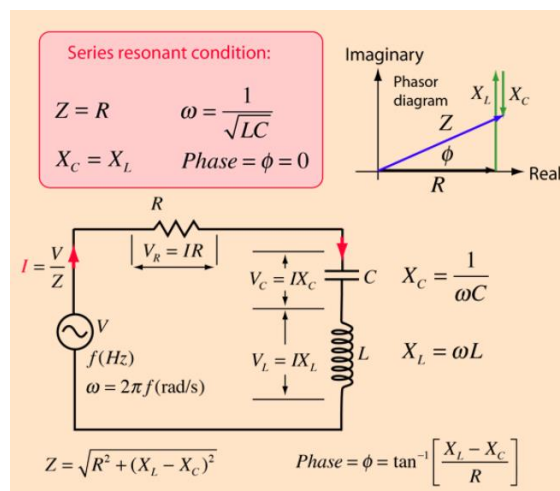


Fig1. The equation for serial RLC impedance calculation

File description

- You will be given RLC_calculator.cpp, component.h, inductor.h, resistor.h and capacitor.h.
- Component class is inherited by resistor, capacitor and inductor classes.
- Be careful, the RLC_calculator.cpp cannot be modified, otherwise you will get 0 points on this program. Other header files can be changed if necessary, but you must still adhere to the guidelines we provide.
- You must submit Makefile, RLC_calculator.cpp, component.h, inductor.h, resistor.h and capacitor.h these five files for this problem; however, if you have another program file, you may compress it into the same directory. In this directory, your code will be compiled and executed.

- (1) RLC_calculator.cpp: main function to implement a series RLC calculator function (**you cannot modify the program in main function**)
- (2) component.h: a class that derived by class capacitor and resistor
- (3) capacitor.h: a class that implement display, operator|| (parallel) and operator+ (series)
- (4) resistor.h/inductor.h: same as capacitor

- ✓ Please write every classes in separate files and write a “Makefile” in your directory to compile your code. Your code can be compiled by typing “make”, and the name of your program should be hw2_RLC.
- ✓ Here are the class template and the output.
- ✓ RLC_calculator.cpp

```
#include <cmath> //for M_PI, sqrt
#include <iomanip> // for setprecision(6)
#include "component.h"
#include "capacitor.h"
#include "resistor.h"
#include "inductor.h"

using namespace std;
double impedance_RLC(resistor r, inductor l, capacitor c,
double f);

int main(int argc, char *argv[])
{
    double f = 10;
    resistor r1(10), r2(32.5);
    cout << "r1: " << r1 << " r2: " << r2 << endl;
    r1 = r1 + r2;
    cout << "r1: " << r1 << " r2: " << r2 << endl;
    r1 = r1 || r2;
    cout << "r1: " << r1 << " r2: " << r2 << endl;

    capacitor c1(10, f), c2(30.5, f);
    cout << "c1: " << c1 << " c2: " << c2 << endl;
    c1 = c1 + c2;
    cout << "c1: " << c1 << " c2: " << c2 << endl;
    c1 = c1 || c2;
    cout << "c1: " << c1 << " c2: " << c2 << endl;

    inductor l1(10.5, f), l2(20.5, f);
    cout << "l1: " << l1 << " l2: " << l2 << endl;
```

```
    l1 = l1 + l2;
    cout << "l1: " << l1 << " l2: " << l2 << endl;
    l1 = l1 || l2;
    cout << "l1: " << l1 << " l2: " << l2 << endl;

    double result = impedance_RLC(r1, l1, c1, f);
    cout << fixed << setprecision(5);
    cout << "Series RLC Circuit Impedance is " << result
    << " \u03A9" << endl;

    return 0;
}
```

✓ resistor.h

```
#ifndef RESISTOR_H_
#define RESISTOR_H_
#include "component.h"

class resistor : protected component
{
private:
public:
    double get_impedance() const;
    resistor();
    // Add some functions or data members by yourself
};

#endif
```

✓ component.h

```
#ifndef COMP_H_
#define COMP_H_
class component
{
protected:
    double impedance;
    double value;
    double frequency;

public:
    virtual double get_impedance() const = 0;
};

#endif
```

✓ \$ make
\$./hw2_RLC

```
r1: 10.00000 Ω  r2: 32.50000 Ω
r1: 42.50000 Ω  r2: 32.50000 Ω
r1: 18.41667 Ω  r2: 32.50000 Ω
c1: 10.00000 μF  (Xc 1591.54943 Ω)  c2: 30.50000 μF  (Xc
521.81949 Ω)
c1: 7.53086 μF  (Xc 2113.36892 Ω)  c2: 30.50000 μF  (Xc
521.81949 Ω)
c1: 38.03086 μF  (Xc 418.48889 Ω)  c2: 30.50000 μF  (Xc
521.81949 Ω)
l1: 10.50000 mH  (Xl 0.65973 Ω)  l2: 20.50000 mH  (Xl 1.28805 Ω)
l1: 31.00000 mH  (Xl 1.94779 Ω)  l2: 20.50000 mH  (Xl 1.28805 Ω)
l1: 12.33981 mH  (Xl 0.77533 Ω)  l2: 20.50000 mH  (Xl 1.28805 Ω)
Series RLC Circuit Impedance Calculator .....
18.41667 Ω  12.33981 mH  (Xl 0.77533 Ω)  38.03086 μF  (Xc
418.48889 Ω)
Series RLC Circuit Impedance is 418.11935 Ω
```

[Problem 2]: Booking THSR/TRA Ticket System

As a college student, it is quite normal to leave your hometown to study. Therefore, THSR(Taiwan High-Speed Rail) and TRA(Taiwan Railways Administration) are two well-known transportation. So, please implement the booking system by completing the definition of the following classes to support the correct execution of the main program (written in main.cpp).

You should complete each cpp files which TA offers.

In this program, you need to take attention that

1. You only can rewrite the function body.
2. You only write `BookingSystem.cpp`, `train_information.cpp`, `THSR.cpp`, `TRA.cpp` and submit them.
3. We will provide Makefile to compile your cpp file with following command:
 - i. `g++ -c train_information.cpp`
 - ii. `g++ -c THSR.cpp`
 - iii. `g++ -c TRA.cpp`
 - iv. `g++ -c BookingSystem.cpp`
 - v. `g++ -c main.cpp`
 - vi. `g++ -o hw2-2 -g -Wall main.o BookingSystem.o TRA.o THSR.o train_information.o`

Attention: Don't worry about the main.cpp and header files, we will provide them. As a result, you only need to ensure that your cpp files can compile with the above command we provide and generate the hw2-2 executable file.

4. The header and main.cpp files are stored in `/home/share/hw2/BookingSystem`.
5. The executable file named `hw2-2` is stored in `/home/share/hw2/BookingSystem/testcase`. You can refer it to complete this problem.
6. The input file `in1` is also stored in `/home/share/hw2/BookingSystem/testcase`. You can refer it to complete this problem.

Main.cpp

```
#include "BookingSystem.h"

using namespace std;

int main()
{
    BookingSystem Myticket;

    Myticket.BookingSystemShell();

    return 0;
}
```

Class description:

train_information.h is shown as below:

```
class train_information
{
private:
    int train_no;
    int drive;
    string date;
    string travel;
    string from, to;

public:
    train_information();
    train_information(int, int, string, string, string, string);
    virtual void print_information();
};
```

Class train_information stores the variables train_no, driving time, date, travel, from, and to. It is a parent class for other classes.

Function print_Information() is the virtual function that prints the information that each class has its own-defined format.

THSR.h is shown as below:

```
class THSR : public train_information
{
private:
    string Class;
    int car;
    string seat;

public:
    THSR();
    THSR(int, int, string, string, string, string, string, int, string);
    void print_information();
};
```

Variable Class stores your preferred class.

Variable car stores the number of which train cabin you sit.

Variable seat stores your seat number in the car.

Function print_Information() prints the information that each class has its own-defined format.

TRA.h is shown as below:

```
class TRA : public train_information
{
private:
    string train_type;
    int car;
    string seat;

public:
    TRA();
    TRA(int, int, string, string, string, string, string, int, string);
    void print_information();
};
```

Variable `train_type` stores which train type you want to travel by.

Variable `car` stores the number of which train cabin you sit.

Variable `seat` stores your seat number in the car.

Function `print_Information()` prints the information that each class has its own-defined format.

BookingSystem.h is shown as below:

```
class BookingSystem
{
private:
    map<string, THSR> THSR_m;
    map<string, TRA> TRA_m;

public:
    BookingSystem();
    void BookingSystemShell();
    void p();
    void b();
    void s();
    void u();
    void parseTicket(string line, int select);
};
```

Variable `THSR_m` is a `map<string, THSR>` data structure, dynamically storing all of your THSR tickets. The inserting map key would be date(string) and second value would be the THSR class object.

Variable `TRA_m` is a `map<string, TRA>` data structure, dynamically storing all of your TRA tickets. The inserting map key would be date(string) and second value would be the TRA class object.

Function `BookingSystemShell()` is the shell that continuously provides services with the prefix "\$ " for the user till typing q(quit).

- i. Users can type in p(print all of my tickets), and `BookingSystemShell()` will call the member function `p()` to complete it. The output permutation is related to the tickets' date(string).
- ii. Users can type in b(buy the tickets), and `BookingSystemShell()` will call the member function `b()` and request entering the corresponding ticket information to complete it. Be careful to handle Round Trip tickets, it will simultaneously add two tickets at one time.
- iii. Users can type in s(search my tickets), and `BookingSystemShell()` will call the member function `s()` and request entering the ticket information about the date to search if you have corresponding THSR/ TRA tickets.
- iv. Users can type in u(use my tickets), and `BookingSystemShell()` will call the member function `u()` and request entering the corresponding ticket information to complete it.

The following are the output of the make command and executing `BookingSystem`.

```
$ make
g++ -c main.cpp
g++ -c BookingSystem.cpp
g++ -c TRA.cpp
g++ -c THSR.cpp
g++ -c train_information.cpp
g++ -o hw2-2 -g -Wall main.o BookingSystem.o TRA.o THSR.o
train_information.o
rm *.o
$
```

```
$ ./hw2-2
====My tickets folder=====
THSR tickets num: 0
TRA tickets num: 0

$: b
THSR
252, 30mins, 2022/06/25 00:00, One way, Hsinchu, Taichung, Standard, 4,
3E
$: b
TRA
3310, 67mins, 2022/06/20 00:00, Round Trip, Taichung, Hsinchu, T.C. Ltd.,
1, 20, 9510, 50mins, 2022/06/23 08:00, Puyuma, 10, 19
$: b
THSR
```



```
27, 40mins, 2022/07/20 12:30, Round Trip, Nangang, Tainan, Business, 6,
20D, 54, 40mins, 2022/07/21 23:30, Business, 6, 5A
$: b
TRA
3310, 67mins, 2022/06/25 00:00, One way, Taichung, Hsinchu, Taroko, 1, 1
$: p
=====My tickets folder=====
THSR tickets num: 3
TRA tickets num: 3

THSR:
ticket#0
    Date: 2022/06/25 00:00
    Train no: 252
    From Hsinchu to Taichung
    Driving time(mins): 30
    Travel: One way
    Class: Standard
    Car, seat: 4, 3E
ticket#1
    Date: 2022/07/20 12:30
    Train no: 27
    From Nangang to Tainan
    Driving time(mins): 40
    Travel: Round Trip
    Class: Business
    Car, seat: 6, 20D
ticket#2
    Date: 2022/07/21 23:30
    Train no: 54
    From Tainan to Nangang
    Driving time(mins): 40
    Travel: Round Trip
    Class: Business
    Car, seat: 6, 5A
TRA:
ticket#0
    Date: 2022/06/20 00:00
    Train no: 3310
    From Taichung to Hsinchu
    Driving time(mins): 67
    Travel: Round Trip
    Train type: T.C. Ltd.
    Car, seat: 1, 20
ticket#1
    Date: 2022/06/23 08:00
    Train no: 9510
    From Hsinchu to Taichung
    Driving time(mins): 50
```

```
Travel: Round Trip
Train type: Puyuma
Car, seat: 10, 19
ticket#2
Date: 2022/06/25 00:00
Train no: 3310
From Taichung to Hsinchu
Driving time(mins): 67
Travel: One way
Train type: Taroko
Car, seat: 1, 1
$: s
2022/06/25 00:00
You have one THSR ticket:
Date: 2022/06/25 00:00
Train no: 252
From Hsinchu to Taichung
Driving time(mins): 30
Travel: One way
Class: Standard
Car, seat: 4, 3E
You have one TRA ticket:
Date: 2022/06/25 00:00
Train no: 3310
From Taichung to Hsinchu
Driving time(mins): 67
Travel: One way
Train type: Taroko
Car, seat: 1, 1
$: s
2022/07/20 12:31
You don't have any ticket whose Date is 2022/07/20 12:31.
$: u
TRA
2022/06/25 00:00
Used.
$: u
THSR
2021/07/20 12:00
You don't have THSR 2021/07/20 12:00 ticket.
$: u
TRA
2022/06/25 00:01
You don't have TRA 2022/06/25 00:01 ticket.
$: u
THSR
2022/07/20 12:30
Used.
$: p
```

```
=====My tickets folder=====
THSR tickets num: 2
TRA tickets num: 2

THSR:
ticket#0
    Date: 2022/06/25 00:00
    Train no: 252
    From Hsinchu to Taichung
    Driving time(mins): 30
    Travel: One way
    Class: Standard
    Car, seat: 4, 3E
ticket#1
    Date: 2022/07/21 23:30
    Train no: 54
    From Tainan to Nangang
    Driving time(mins): 40
    Travel: Round Trip
    Class: Business
    Car, seat: 6, 5A

TRA:
ticket#0
    Date: 2022/06/20 00:00
    Train no: 3310
    From Taichung to Hsinchu
    Driving time(mins): 67
    Travel: Round Trip
    Train type: T.C. Ltd.
    Car, seat: 1, 20
ticket#1
    Date: 2022/06/23 08:00
    Train no: 9510
    From Hsinchu to Taichung
    Driving time(mins): 50
    Travel: Round Trip
    Train type: Puyuma
    Car, seat: 10, 19

$: q

exit.
```