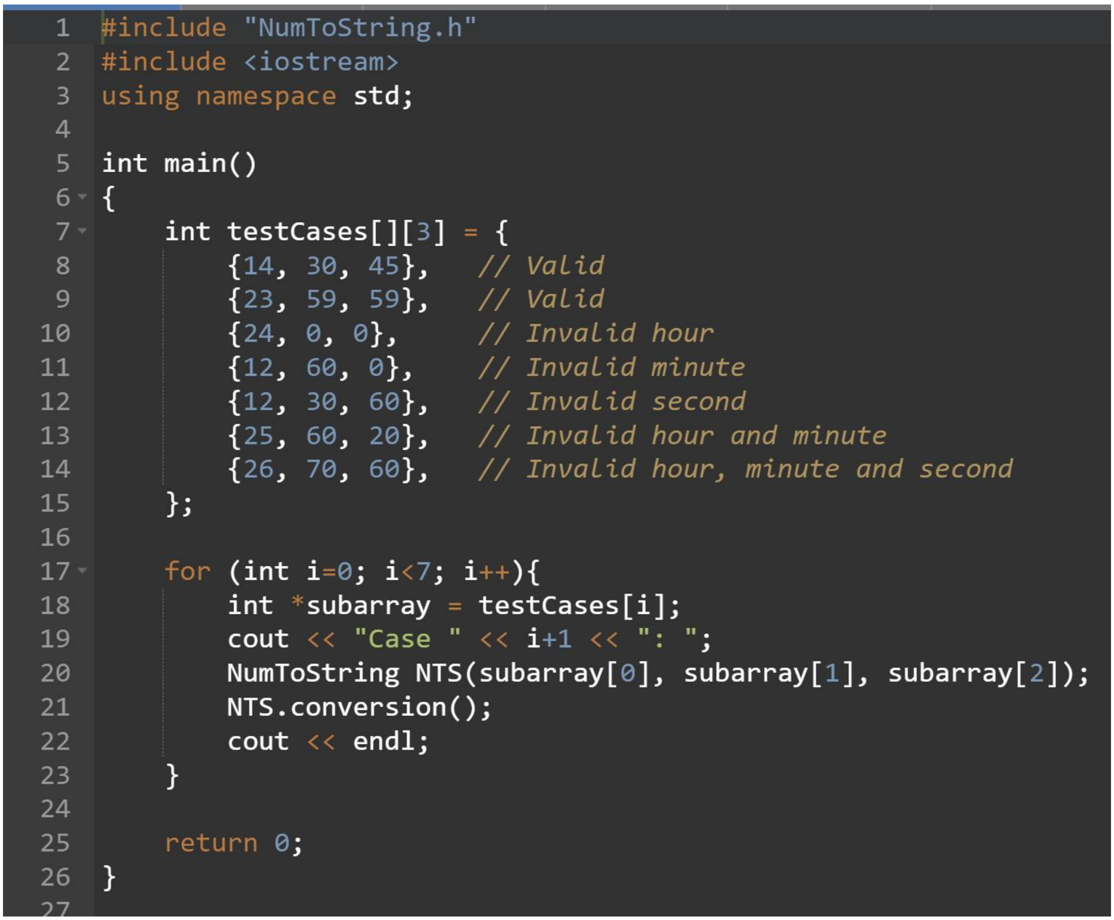
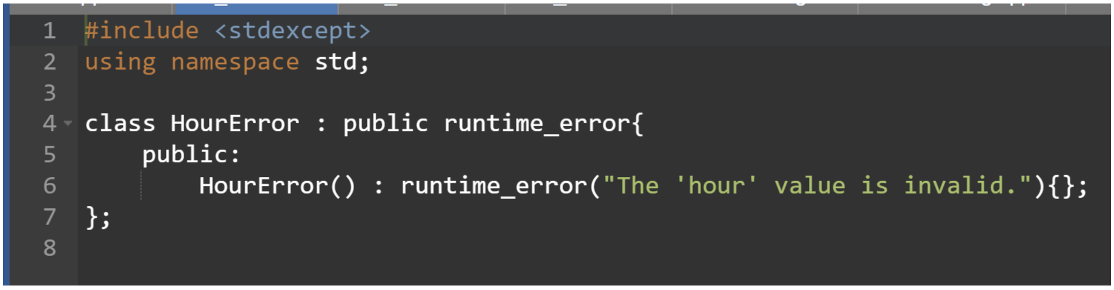
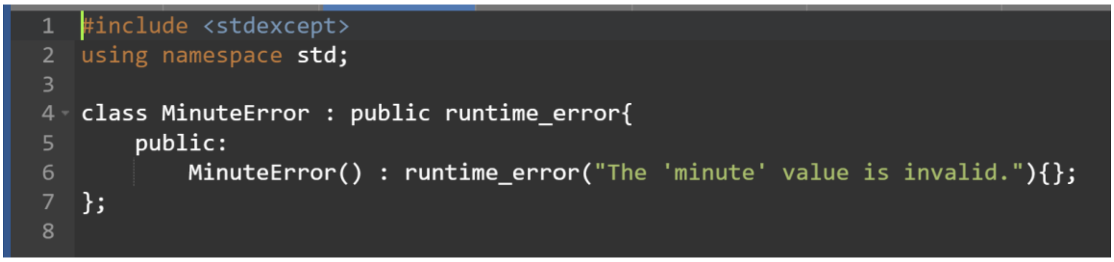


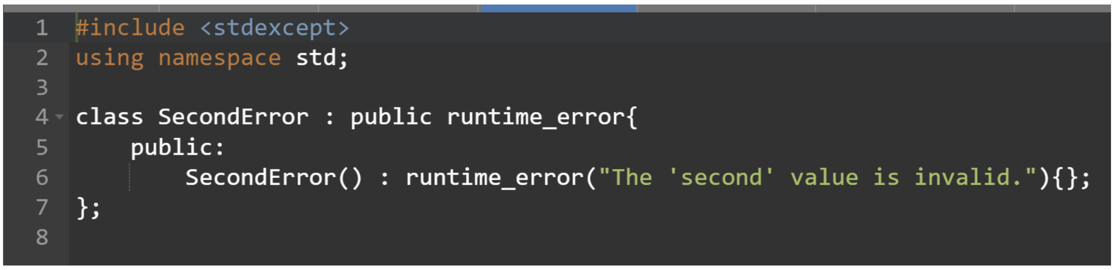
```cpp
#include "NumToString.h"
#include <iostream>
using namespace std;

int main()
{
    int testCases[][3] = {
        {14, 30, 45},   // Valid
        {23, 59, 59},   // Valid
        {24, 0, 0},     // Invalid hour
        {12, 60, 0},    // Invalid minute
        {12, 30, 60},   // Invalid second
        {25, 60, 20},   // Invalid hour and minute
        {26, 70, 60},   // Invalid hour, minute and second
    };

    for (int i=0; i<7; i++){
        int *subarray = testCases[i];
        cout << "Case " << i+1 << ": ";
        NumToString NTS(subarray[0], subarray[1], subarray[2]);
        NTS.conversion();
        cout << endl;
    }

    return 0;
}
```



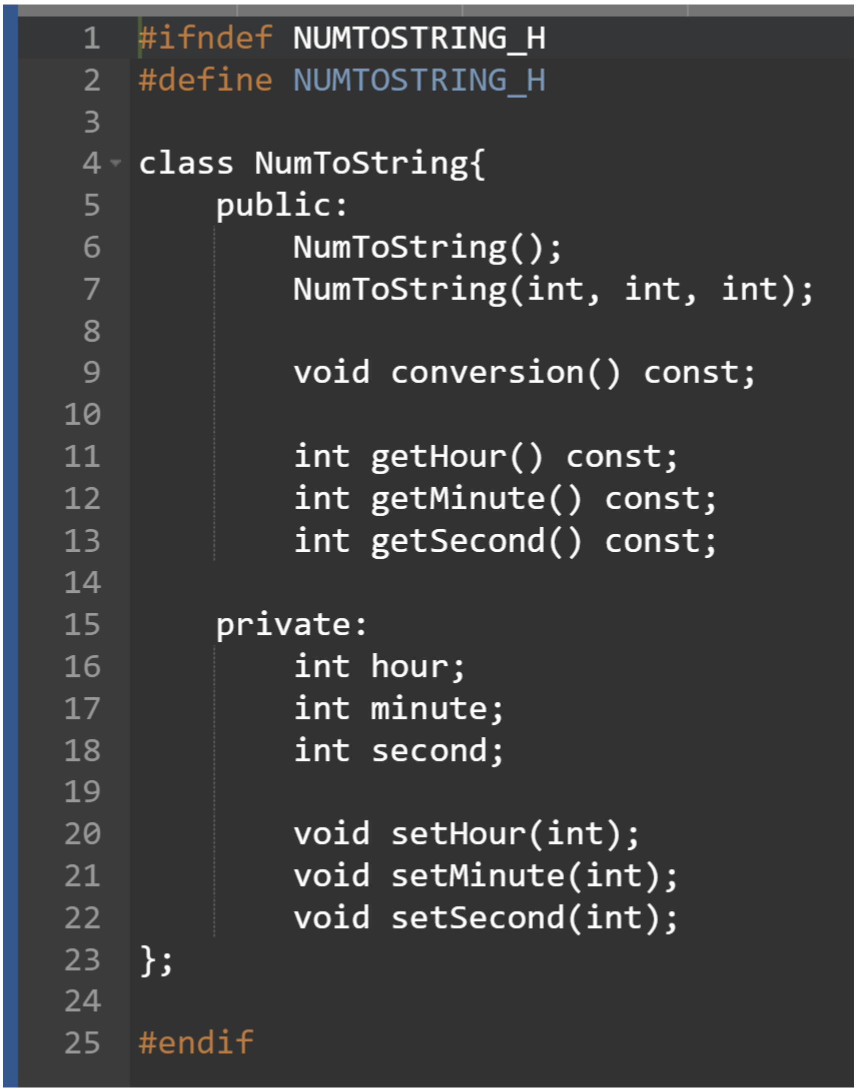
```cpp
#include <stdexcept>
using namespace std;

class HourError : public runtime_error{
    public:
        HourError() : runtime_error("The 'hour' value is invalid."){};
};
```



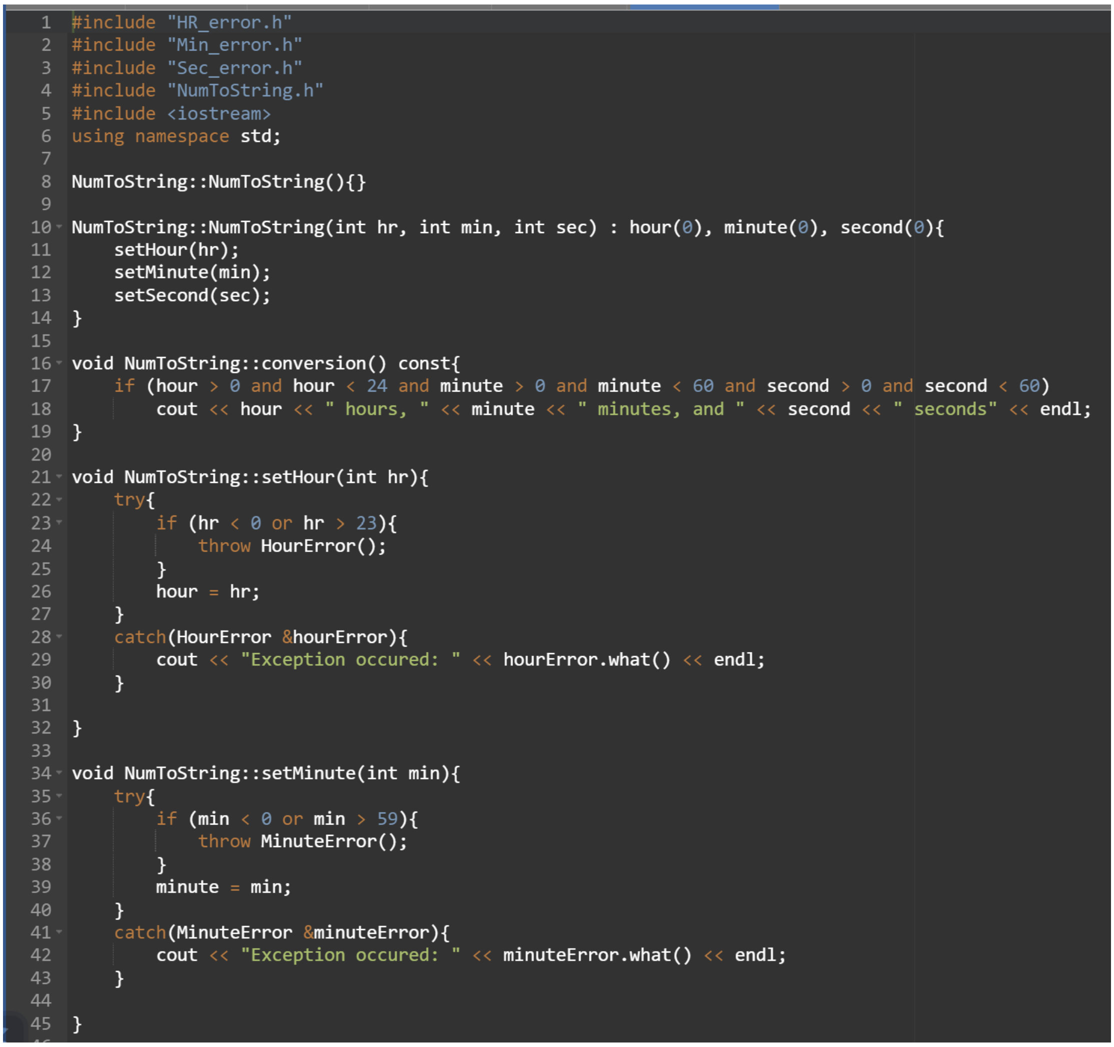
```cpp
#include <stdexcept>
using namespace std;

class MinuteError : public runtime_error{
    public:
        MinuteError() : runtime_error("The 'minute' value is invalid."){};
};
```



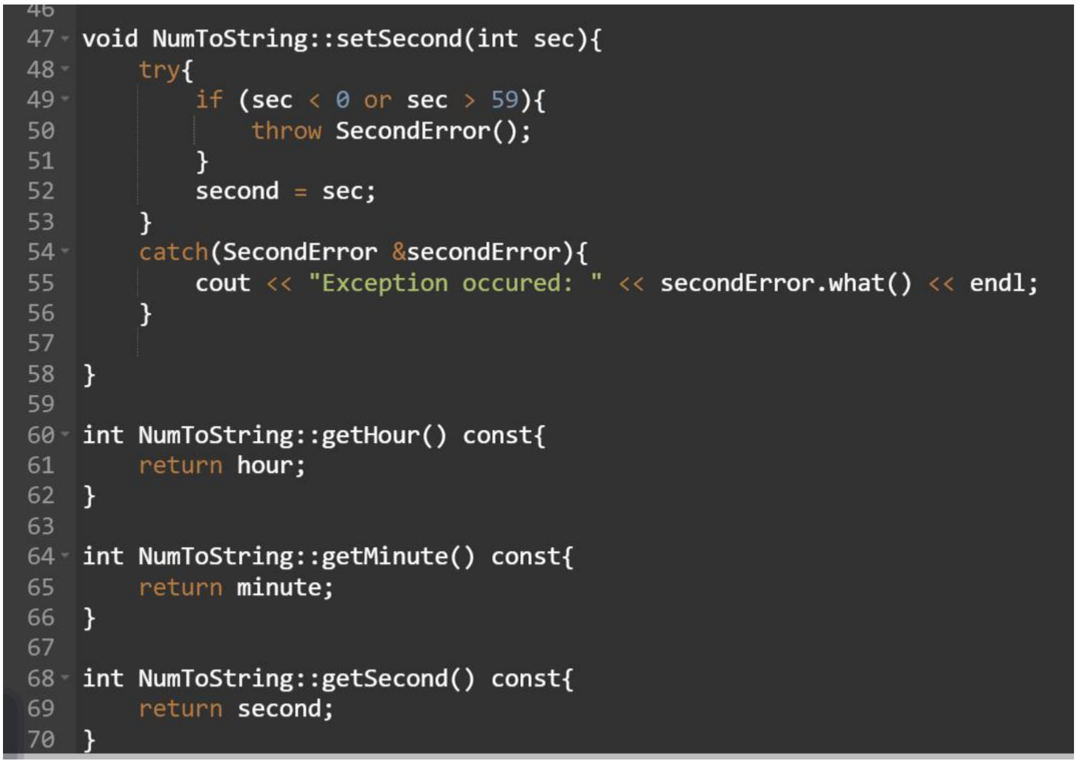
```cpp
#include <stdexcept>
using namespace std;

class SecondError : public runtime_error{
    public:
        SecondError() : runtime_error("The 'second' value is invalid."){};
};
```



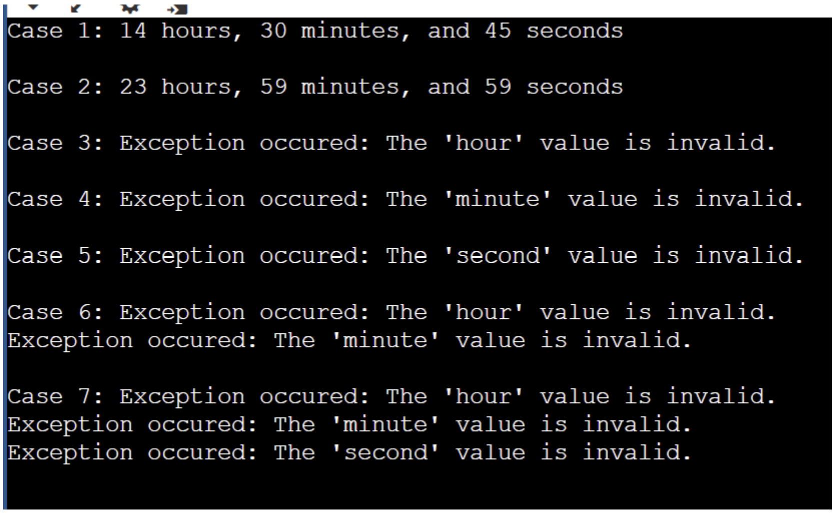
```cpp
#ifndef NUMTOSTRING_H
#define NUMTOSTRING_H

class NumToString{
    public:
        NumToString();
        NumToString(int, int, int);

        void conversion() const;

        int getHour() const;
        int getMinute() const;
        int getSecond() const;

    private:
        int hour;
        int minute;
        int second;

        void setHour(int);
        void setMinute(int);
        void setSecond(int);
};

#endif
```



```cpp
#include "HR_error.h"
#include "Min_error.h"
#include "Sec_error.h"
#include "NumToString.h"
#include <iostream>
using namespace std;

NumToString::NumToString(){}

NumToString::NumToString(int hr, int min, int sec) : hour(0), minute(0), second(0){
    setHour(hr);
    setMinute(min);
    setSecond(sec);
}

void NumToString::conversion() const{
    if (hour > 0 and hour < 24 and minute > 0 and minute < 60 and second > 0 and second < 60)
        cout << hour << " hours, " << minute << " minutes, and " << second << " seconds" << endl;
}

void NumToString::setHour(int hr){
    try{
        if (hr < 0 or hr > 23){
            throw HourError();
        }
        hour = hr;
    }
    catch(HourError &hourError){
        cout << "Exception occured: " << hourError.what() << endl;
    }

}

void NumToString::setMinute(int min){
    try{
        if (min < 0 or min > 59){
            throw MinuteError();
        }
        minute = min;
    }
    catch(MinuteError &minuteError){
        cout << "Exception occured: " << minuteError.what() << endl;
    }

}
```



```cpp

void NumToString::setSecond(int sec){
    try{
        if (sec < 0 or sec > 59){
            throw SecondError();
        }
        second = sec;
    }
    catch(SecondError &secondError){
        cout << "Exception occured: " << secondError.what() << endl;
    }

}

int NumToString::getHour() const{
    return hour;
}

int NumToString::getMinute() const{
    return minute;
}

int NumToString::getSecond() const{
    return second;
}
```



```
Case 1: 14 hours, 30 minutes, and 45 seconds

Case 2: 23 hours, 59 minutes, and 59 seconds

Case 3: Exception occured: The 'hour' value is invalid.

Case 4: Exception occured: The 'minute' value is invalid.

Case 5: Exception occured: The 'second' value is invalid.

Case 6: Exception occured: The 'hour' value is invalid.
Exception occured: The 'minute' value is invalid.

Case 7: Exception occured: The 'hour' value is invalid.
Exception occured: The 'minute' value is invalid.
Exception occured: The 'second' value is invalid.
```