```
/*
*******************************************************************************
FILENAME        ohmslaw_ver3.cpp

Encoding        UTF-8

DESCRIPTION     Calculate Voltage, Resistance, Current.

FUNCTIONS       Using pointer.

NOTES           Menu language - English

Compiler        g++ 9.3.0 amd64 running @ Ubuntu 20.04 LTS

Lang dialect    ISO C++14 (g++ by default uses option ’-std=gnu++14’)

                Copyright L.Krüger 2020. All rights reserved.

AUTHOR          Leif Krüger, leif@leifkruger.se

CHANGES

REF NO  VERSION     DATE (YYMMDD)   WHO DETAIL
-------------------------------------------------------------------------------
        1           2020-11-04      LK  Start date
        2           2020-11-05      LK  Uses more general functions
        3           2020-11-06      LK  Modify error handling "water proof"
*******************************************************************************
*/

#include <iostream>
#include <string>
#include <sstream>
#include <climits>
using namespace std;

void checkInput(string quantity, double *uriVariable);
void showResultat(string quantity, double *uriVariable1, double *uriVariable2);

//Struct for U=R*I
struct ohmsLaw {
    double voltage;
    double current;
    double resistance;
};

int main() {
    char chooseRunagain;
    do {
        string selectCalc; //Use a string for error handling
        char selectedUri;
        ohmsLaw uri;
        cout << "\nOhm’s law U=R*I" << endl;
        cout << "===============" << endl;
        cout << "Select the quantity to be calculated:" << endl;
        cout << "Voltage (u), Resistance (r), Current (i), or Quit (q)? ";
        getline(cin, selectCalc); //Read string for error handling
        stringstream(selectCalc) >> selectedUri; //Only use first character
        selectedUri = tolower(selectedUri);

        if (selectedUri == ’u’) {
            checkInput("Current (A)", &uri.current);
            checkInput("Resistance (\u2126)", &uri.resistance);
            showResultat("Voltage", &uri.current, &uri.resistance);
        }
        else if (selectedUri == ’r’) {
            checkInput("Voltage (V)", &uri.voltage);
            checkInput("Current (A)", &uri.current);
            showResultat("Resistance", &uri.voltage, &uri.current);
        }
        else if (selectedUri == ’i’) {
            checkInput("Voltage (V)", &uri.voltage);
            checkInput("Resistance (\u2126)", &uri.resistance);
            showResultat("Current", &uri.voltage, &uri.resistance);
        }
        else if (selectedUri == ’q’) {
            chooseRunagain = ’n’;
        }
        else {
            cout << "\nSorry, wrong menu selection!\n";
        }
    } while (chooseRunagain != ’n’);
    return 0;
}

//Function for input control with error handling
void checkInput(string quantity, double *uriVariable) {
    do {
        string testString;
        cout << quantity << "? ";
        getline (cin, testString);
        stringstream(testString) >> *uriVariable;
        if (*uriVariable == 0) {
            cout << "Please check the entry. Try again! " << endl;
        }
    }
    while (*uriVariable == 0);
}

//Function for output of result
void showResultat(string quantity, double *uriVariable1, double *uriVariable2) {
    cout << "\nFormula: U=R*I" << endl;
    if (quantity == "Voltage") {
        cout << "Known: Current " << *uriVariable1 << " A * Resistance "
         << *uriVariable2 << " \u2126 " << endl;
        cout << "Result: " << quantity << " = " << *uriVariable1 * *uriVariable2
         << " V" << endl;
    }
    else if (quantity == "Resistance") {
        cout << "Known: Voltage " << *uriVariable1 << " V / Current "
         << *uriVariable2 << " A " << endl;
        cout << "Result: " << quantity << " = " << *uriVariable1 / *uriVariable2
         << " \u2126 " << endl;
    }
    else if (quantity == "Current") {
        cout << "Known: Voltage " << *uriVariable1 << " V / Resistance "
         << *uriVariable2 << " \u2126 " << endl;
        cout << "Result: " << quantity << " = " << *uriVariable1 / *uriVariable2
         << " A" << endl;
    }
}
```