A picture containing logo

Description automatically generated

**Homework # 1**

**01286131 Object-oriented Programming**

**Software Engineering Program**

**Faculty of Engineering, KMITL**

By

65011693 Soe Moe Htet

No.1.1

#include <iostream>

using namespace std;

void print (string star, string stars, string spaces,

            string war\_sp, string mag\_sp, string nin\_sp, string fig\_sp,

            string warrior, string mage, string ninja, string fighter)

{

    // First row

    cout << stars << endl;

    // Second row

    cout << star << spaces << star << spaces  << star << endl;

    // Third row

    cout << "\* Warrior: " << war\_sp << warrior << " \*";

    // Fourth row

    cout << " Mage:    " << mag\_sp << mage << " \*" << endl;

    // Fifth row

    cout << star << spaces << star << spaces  << star << endl;

    // Sixth row

    cout << stars << endl;

    // Seventh row

    cout << star << spaces << star << spaces  << star << endl;

    // Eighth row

    cout << "\* Ninja:   " << nin\_sp << ninja << " \*";

    // Ninth row

    cout << " Fighter: " << fig\_sp << fighter << " \*" << endl;

    // Tenth row

    cout << star << spaces << star << spaces  << star << endl;

    // 11th row

    cout << stars << endl;

}

int main()

{

    string warrior, mage, ninja, fighter;

    cout << "Enter Warrior name: ";

    cin  >> warrior;

    cout << "Enter Mage name: ";

    cin  >> mage;

    cout << "Enter Ninja name: ";

    cin  >> ninja;

    cout << "Enter Fighter name: ";

    cin  >> fighter;

    int warr\_len = warrior.length();

    int mag\_len = mage.length();

    int ninja\_len = ninja.length();

    int fighter\_len = fighter.length();

    int highest\_len = warr\_len;

    if (highest\_len < mag\_len)

    {

        highest\_len = mag\_len;

    }

    else if (highest\_len < ninja\_len)

    {

        highest\_len = ninja\_len;

    }

    else if (highest\_len < fighter\_len)

    {

        highest\_len = fighter\_len;

    }

    else{

        highest\_len = warr\_len;

    }

    int stars\_len = ((12 + highest\_len) \* 2 ) + 1;

    string stars(stars\_len, '\*');

    string spaces( ((stars\_len / 2)-1) , ' ');

    int war = highest\_len - warr\_len;

    int mag = highest\_len - mag\_len;

    int nin = highest\_len - ninja\_len;

    int fig = highest\_len - fighter\_len;

    string star = "\*";

    string war\_sp(war, ' ');

    string mag\_sp(mag, ' ');

    string nin\_sp(nin, ' ');

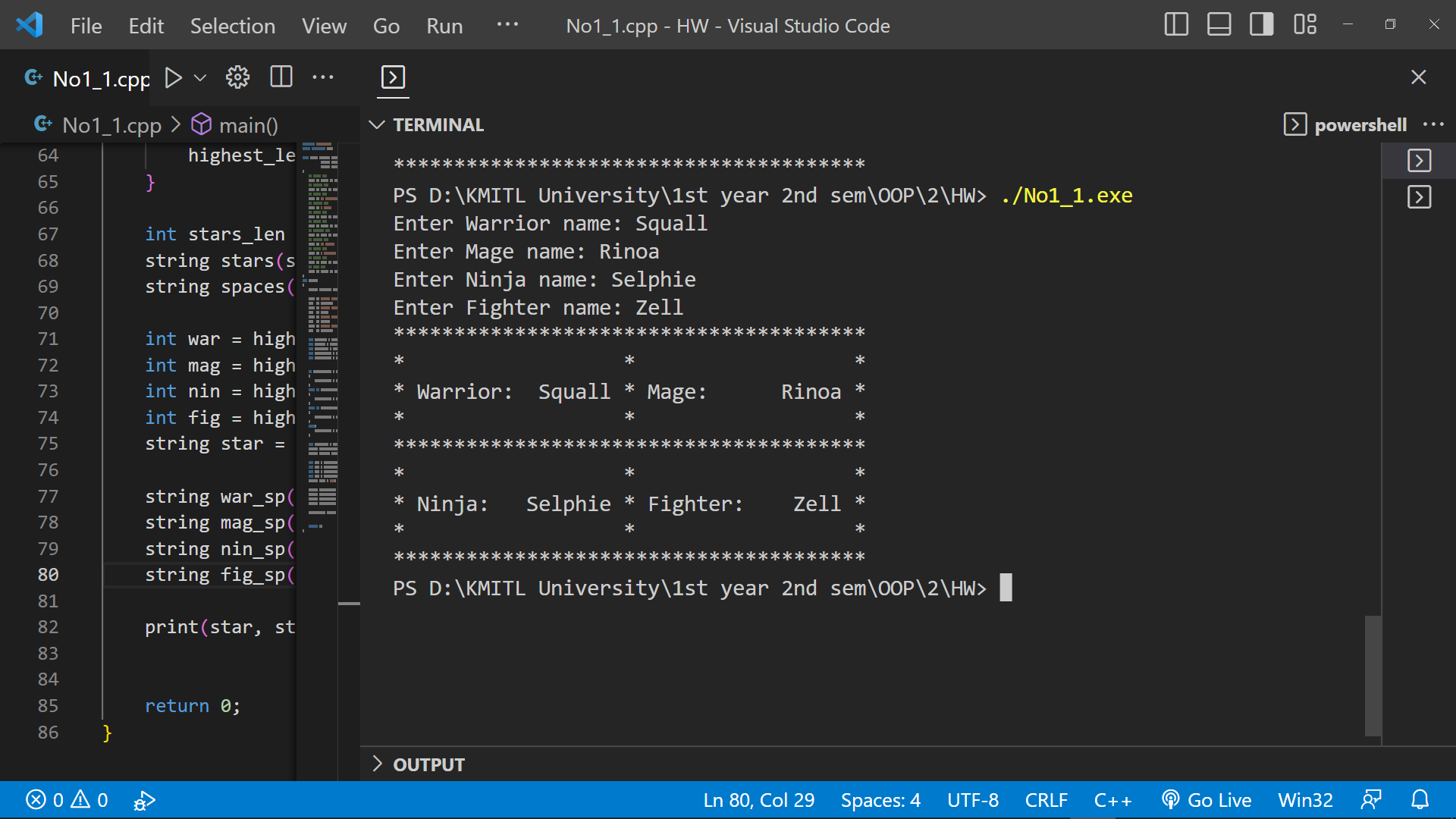
    string fig\_sp(fig, ' ');

    print(star, stars, spaces, war\_sp, mag\_sp, nin\_sp, fig\_sp, warrior, mage, ninja, fighter);

    return 0;

}

Result



No1.2

#include <iostream>

using namespace std;

void print (string plus, string single, string equals, string spaces, string minus,

            string war\_sp, string mag\_sp, string nin\_sp, string fig\_sp,

            string warrior, string mage, string ninja, string fighter)

{

    // First row

    cout << plus << equals << plus << equals << plus << endl;

    // Second row

    cout << single << spaces << single << spaces  << single << endl;

    // Third row

    cout << "| warrior: " << war\_sp << warrior << " |";

    // Fourth row

    cout << " Mage:    " << mag\_sp << mage << " |" << endl;

    // Fifth row

    cout << single << spaces << single << spaces  << single << endl;

    // Sixth row

    cout << plus << minus << plus << minus << plus << endl;

    // Seventh row

    cout << single << spaces << single << spaces  << single << endl;

    // Eighth row

    cout << "| Ninja:   " << nin\_sp << ninja << " |";

    // Ninth row

    cout << " Fighter: " << fig\_sp << fighter << " |" << endl;

    // Tenth row

    cout << single << spaces << single << spaces  << single << endl;

    // 11th row

    cout << plus << equals << plus << equals << plus << endl;

}

int main() {

    string warrior, mage, ninja, fighter;

    cout << "Enter Warrior name: ";

    cin  >> warrior;

    cout << "Enter Mage name: ";

    cin  >> mage;

    cout << "Enter Ninja name: ";

    cin  >> ninja;

    cout << "Enter Fighter name: ";

    cin  >> fighter;

    int warr\_len = warrior.length();

    int mag\_len = mage.length();

    int ninja\_len = ninja.length();

    int fighter\_len = fighter.length();

    int highest\_len = warr\_len;

    if (highest\_len < mag\_len)

    {

        highest\_len = mag\_len;

    }

    else if (highest\_len < ninja\_len)

    {

        highest\_len = ninja\_len;

    }

    else if (highest\_len < fighter\_len)

    {

        highest\_len = fighter\_len;

    }

    else{

        highest\_len = warr\_len;

    }

    int base\_len = ((12 + highest\_len) \* 2 ) + 1;

    string equals(((base\_len / 2)-1), '=');

    string minus(((base\_len / 2)-1), '-');

    string spaces( ((base\_len / 2)-1) , ' ');

    string plus   = "+";

    string single = "|";

    int war = highest\_len - warr\_len;

    int mag = highest\_len - mag\_len;

    int nin = highest\_len - ninja\_len;

    int fig = highest\_len - fighter\_len;

    string war\_sp(war, ' ');

    string mag\_sp(mag, ' ');

    string nin\_sp(nin, ' ');

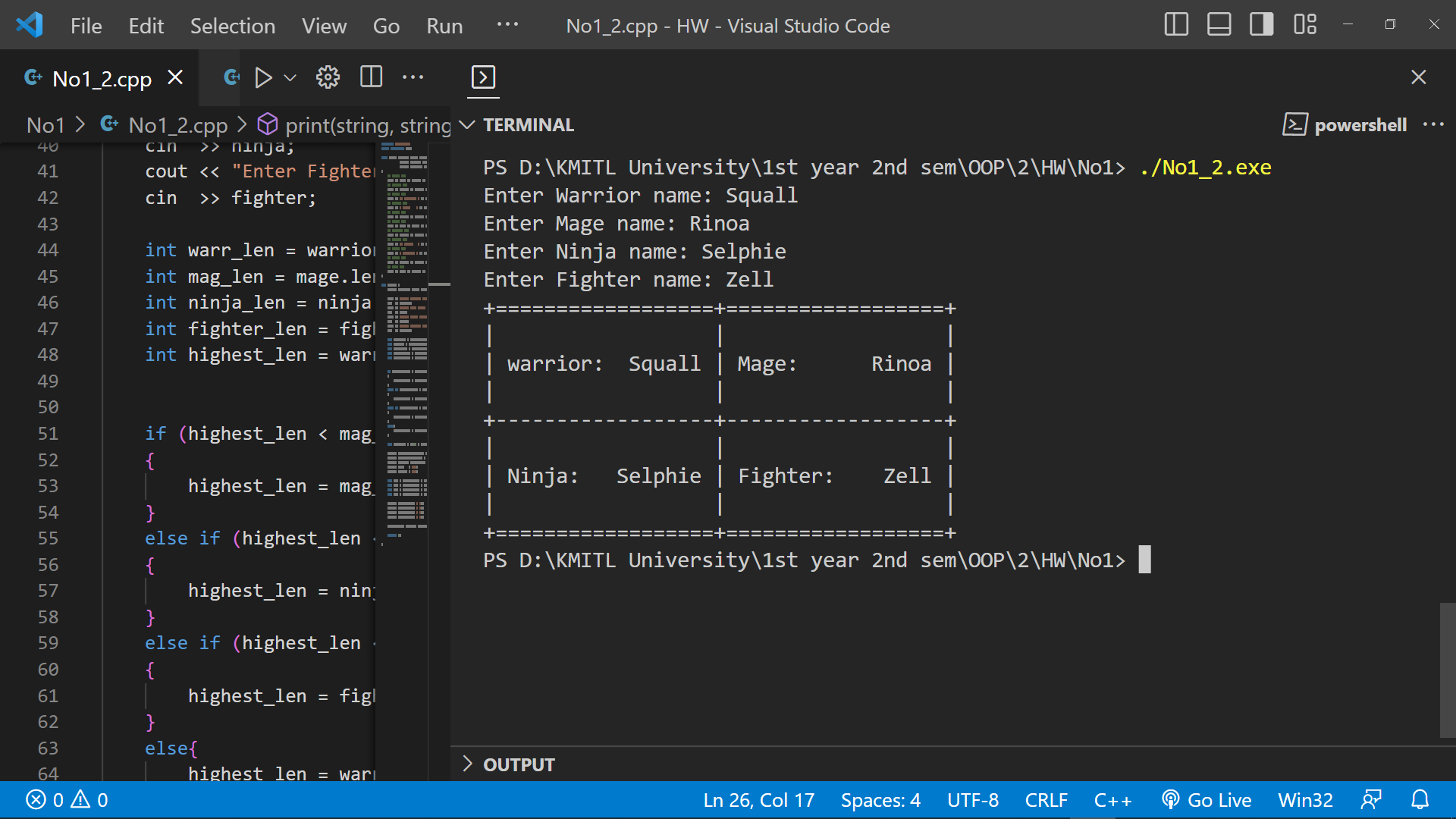
    string fig\_sp(fig, ' ');

    print(plus, single, equals, spaces, minus, war\_sp, mag\_sp, nin\_sp, fig\_sp, warrior, mage, ninja, fighter);

    return 0;

}

Result



No2

No2.1

#include <iostream>

#include <iomanip>

using namespace std;

int main(){

    double temp = 0;

    cout<< "Fahr    Celsius" << endl;

    for (double f= 0.0; f <= 300.0 ; f+=20 )

    {

        temp = (5.0/9.0) \* (f-32.0);

        if (temp > 100)

        {

          cout<< setw(3) << f << setw(12) << setprecision(4) << temp << endl;

        }

        else

        {

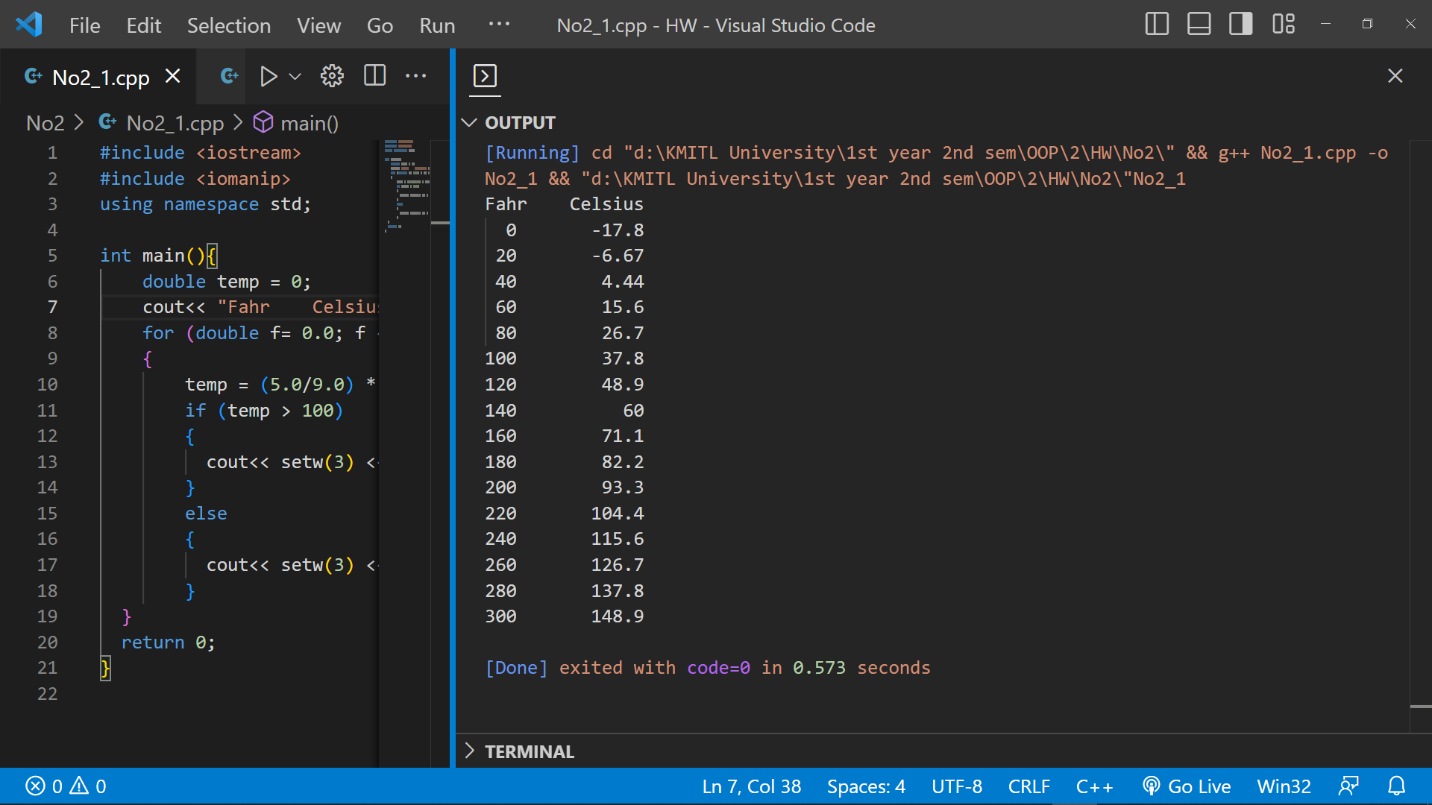
          cout<< setw(3) << f << setw(12) << setprecision(3) << temp << endl;

        }

  }

  return 0;

}



No2.2

#include <iostream>

#include <iomanip>

using namespace std;

int main(){

    double temp = 0;

    cout<< "Fahr    Celsius" << endl;

    for (double f= 0.0; f <= 300.0 ; f+= 40 )

    {

        temp = (5.0/9.0) \* (f-32.0);

        if (temp > 100)

        {

          cout<< setw(3) << f << setw(12) << setprecision(4) << temp << endl;

        }

        else

        {

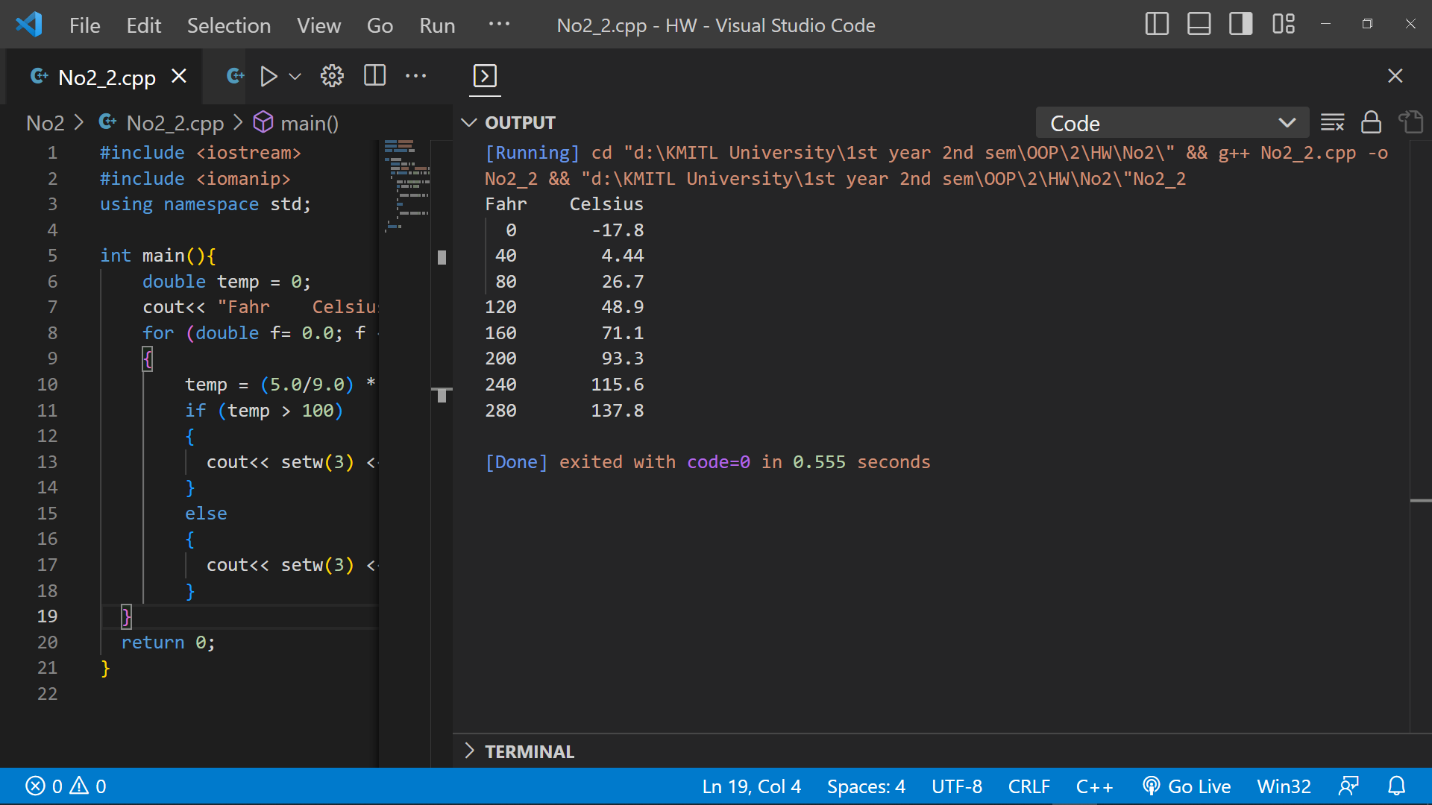
          cout<< setw(3) << f << setw(12) << setprecision(3) << temp << endl;

        }

  }

  return 0;

}



No2.3

#include <iostream>

#include <iomanip>

using namespace std;

int main(){

    double temp = 0;

    cout<< "Fahr    Celsius" << endl;

    for (double f= 300.0; f >= 0.0 ; f-= 20 )

    {

        temp = (5.0/9.0) \* (f-32.0);

        if (temp > 100)

        {

          cout<< setw(3) << f << setw(12) << setprecision(4) << temp << endl;

        }

        else

        {

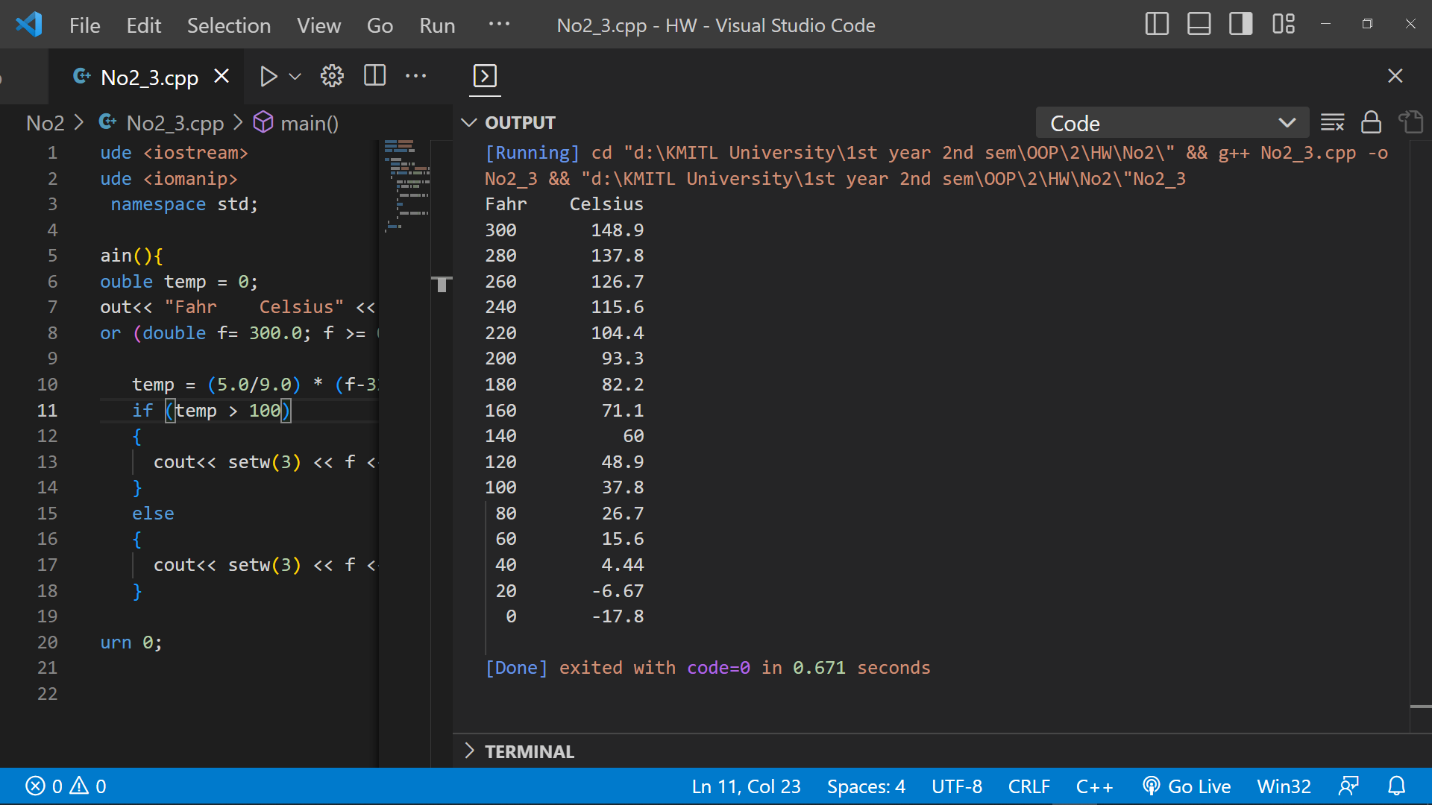
          cout<< setw(3) << f << setw(12) << setprecision(3) << temp << endl;

        }

  }

  return 0;

}



No3

//// random.hpp

#ifndef MY\_RANDOM\_HPP

#define MY\_RANDOM\_HPP

#include <random>

class Rand\_double {

public:

    using seed\_type = std::random\_device::result\_type;

    Rand\_double(double low, double high): dist{low,high} {}

    // draw an integer number

    double operator()() { return dist(re); }

    // choose new random engine seed

    void seed(seed\_type s) { re.seed(s); }

private:

    std::default\_random\_engine re;

    std::uniform\_real\_distribution<double> dist;

};

#include <iomanip>

#include <iostream>

#include <vector>

#include <math.h>

template<typename T\_>

inline constexpr

    T\_ pi\_v{3.141592653589793238462643383279502884L};

inline constexpr double pi = pi\_v<double>;

int main()

{

    constexpr double rnd\_min = 0.0, rnd\_max = 1.0;

    Rand\_double rnd{rnd\_min, rnd\_max};

    std::random\_device rd;

    rnd.seed(rd());

    std::cout << std::fixed << std::setprecision(3);

    std::cout << "Please enter N: ";

    int N;

    std::cin>> N;

    int count = 0;

    double total\_squared\_errors = 0.0;

    while (count != N){

        double random\_num = rnd();

        total\_squared\_errors += (0.5-random\_num)\*(0.5-random\_num);

        count++;

    }

    double mean\_squared\_error = total\_squared\_errors / N;

    std::cout << "MSE is : " << mean\_squared\_error;

    return 0;

}

// end::lab1-3b[]

#endif /\* MY\_RANDOM\_HPP \*/

