Homework # 1

01286131 Object-oriented Programming

Software Engineering Program

Faculty of Engineering, KMITL

By

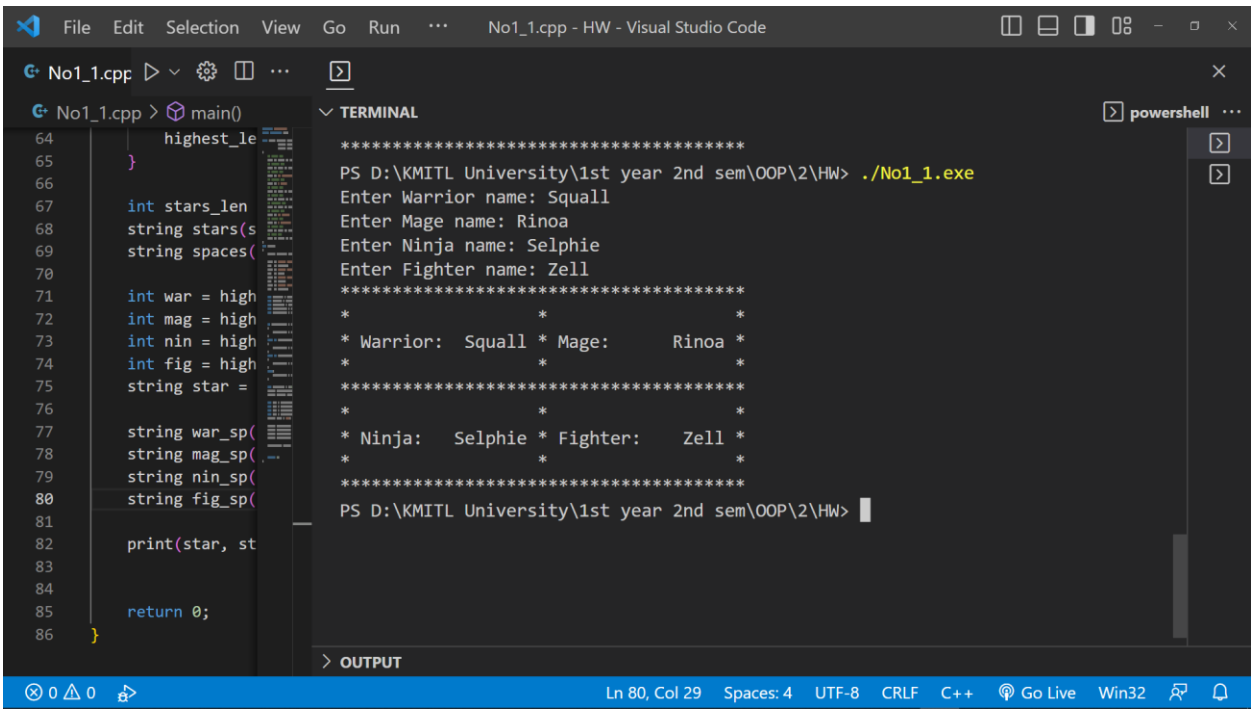
65011693 Soe Moe Htet

No.1.1

```cpp
#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



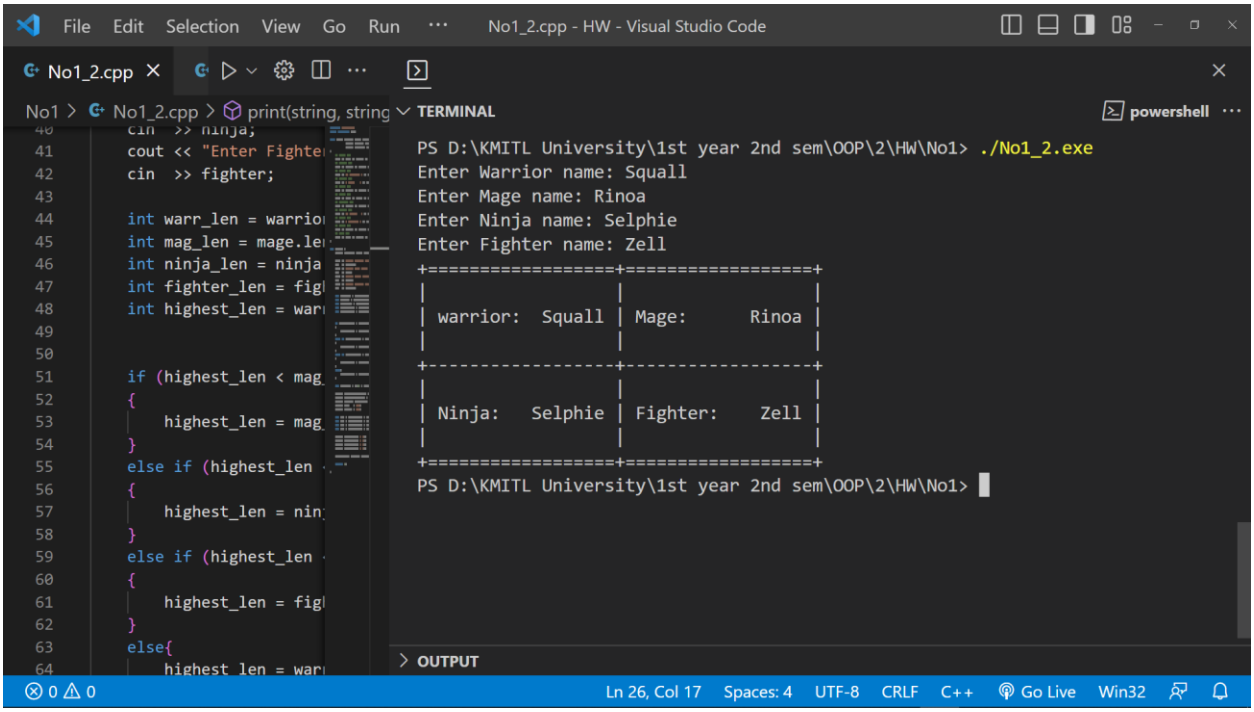
No1.2

```
#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;

}
```

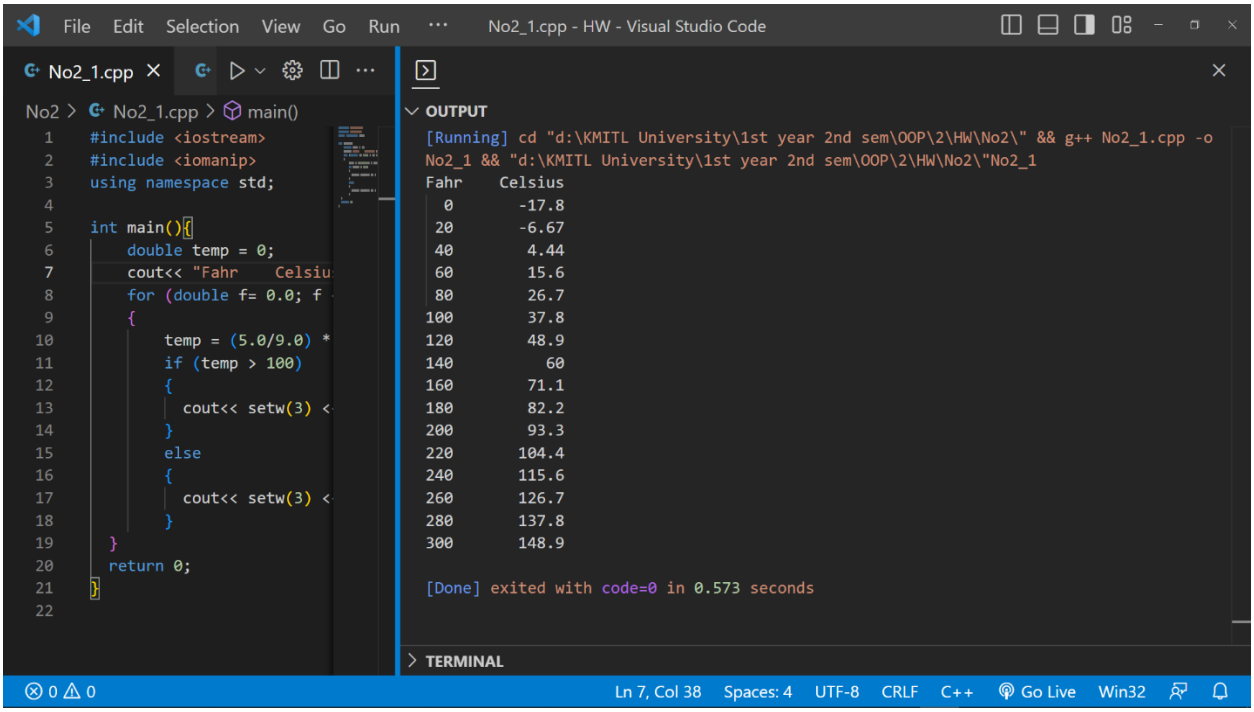
Result



No2

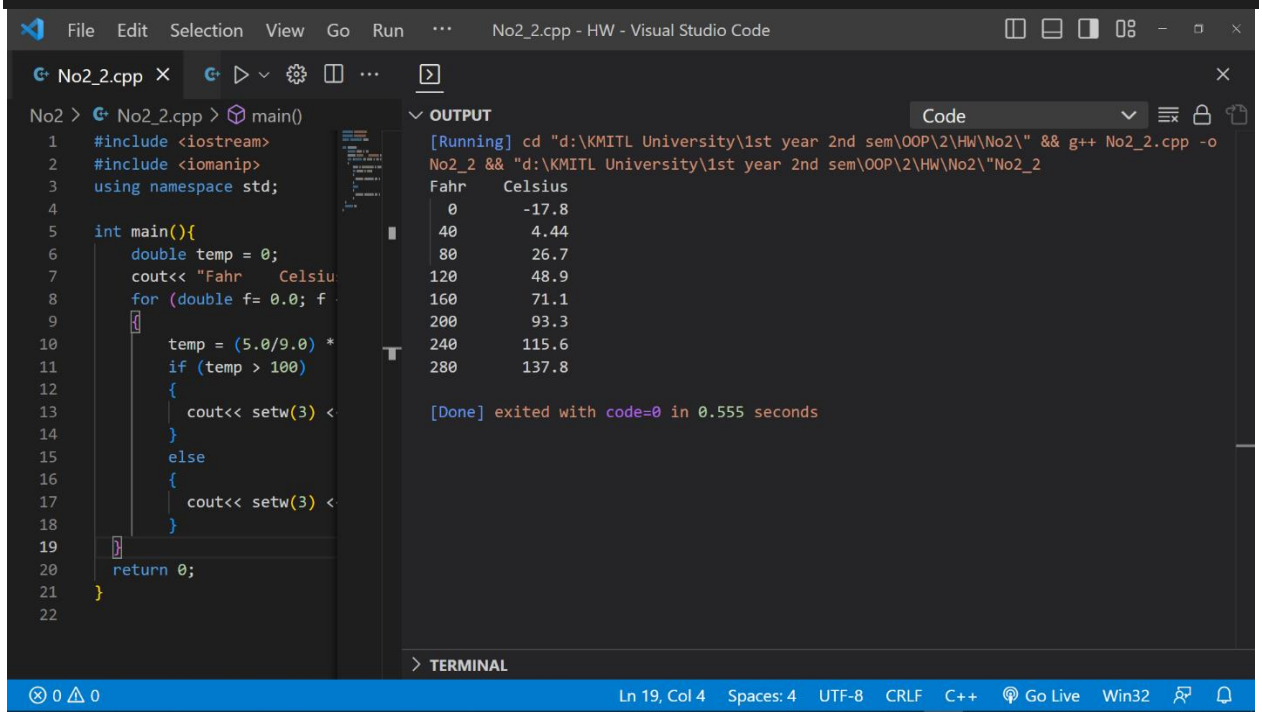
No2.1

```cpp
#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;
}
```


File Edit Selection View Go Run ···
No2_1.cpp - HW - Visual Studio Code
No2_1.cpp
No2 > No2_1.cpp > main()
#include <iostream>
#include <iomanip>
using namespace std;
int main(){
double temp = 0;
cout<< "Fahr Celsiu
for (double f= 0.0; f
{
temp = (5.0/9.0) *
if (temp > 100)
{
cout<< setw(3) <
}
else
{
cout<< setw(3) <
}
}
return 0;
}
OUTPUT
[Running] cd "d:\KMITL University\1st year 2nd sem\OOP\2\HW\No2\" && g++ No2_1.cpp -o No2_1 && "d:\KMITL University\1st year 2nd sem\OOP\2\HW\No2\"No2_1
Fahr Celsius
0 -17.8
20 -6.67
40 4.44
60 15.6
80 26.7
100 37.8
120 48.9
140 60
160 71.1
180 82.2
200 93.3
220 104.4
240 115.6
260 126.7
280 137.8
300 148.9
[Done] exited with code=0 in 0.573 seconds
TERMINAL
0 0
Ln 7, Col 38 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

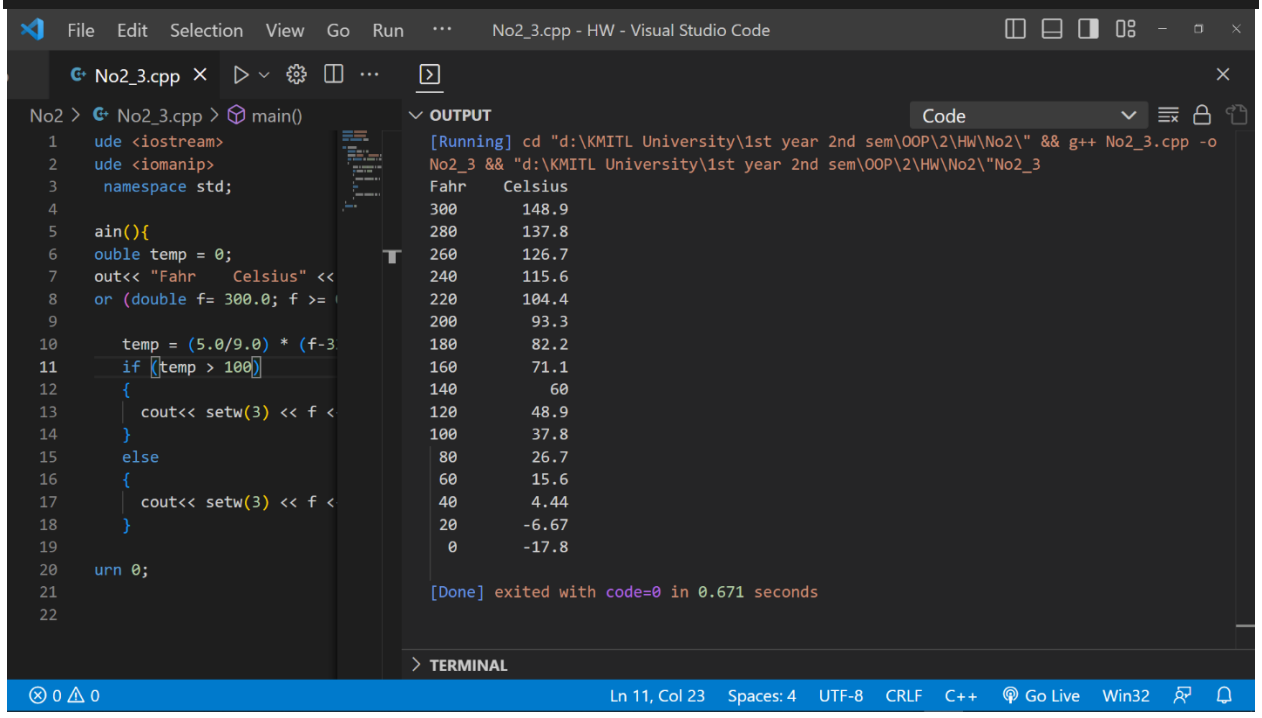
No2.2

```cpp
#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

```cpp
#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

```cpp
//// 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{3.14159265358979323846264338327950288 4L};

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 */
```