

İSTANBUL TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM FAKÜLTESİ
Date: 23.06.2021
Prof.Dr.Hatice Köse
Assoc.Prof.Dr.Gülşen Eryiğit

Student Id	:	
Name Surname	:	
Signature	:	

OBJECT ORIENTED PROGRAMMING
FINAL EXAM

Question #1 (65pts)

Design and develop a program for a war game. Create a class for warlords who has two integer attributes showing how powerful they are in the war arena, such as **power** and **magic**. When two warlords have compared, the sum of their power and magic shows the more powerful one. No warlord can have non-positive magic or negative power. If a warlord object initialized with non-positive magic or negative power, then the program should interrupt/prevent the creation of such an object and warn the user in these exceptional cases: “A warlord cannot be a non-magic wielder!”, “A warlord cannot have negative power!”

In the main function, you should put the given warlords in a container and sort them in descending order. Notice that the comparison function must be available for all sorts of types. You can use STL vector for the container. Use the warlord’s **Print** function to print the magic and power of all warlords after they are sorted.

Download the skeleton code **question1_skeleton.cpp** from Ninova, fill in necessary parts to ensure the requirements listed above. Then, upload your program as **question1.cpp** to the Ninova system. You shouldn’t remove or change the given codes in the **question1_skeleton.cpp** file, but you should add your codes to appropriate locations.

Attention: You can use your course slides. STL is **allowed only for the container, iterator, and algorithm**.

Getting help from any other person, or from the internet is **not allowed, and will be punished**.

When you fill the given code as expected, you should get the following sample output:

```

#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

class Warlord {
private:
    int power;
    int magic;
public:
    /* Write a constructor */

    /* Write Print function */

};

/* Write a comparison function which will return true if first parameter is bigger than the second */

/* Use typedef for your container to work with the code */
typedef Container::iterator Iterator;

/* You should not change or remove the given code parts. Add necessary codes to get the same result with sample output */
int main() {

    Warlord w1(1, 1);
    Warlord w2(2, 2);
    Warlord w3(2, 1);

    /* Create a container and put the w1,w2, and w3 into container */

    cout << "Unsorted Warlords" << endl;
    for (Iterator i = lords.begin(); i != lords.end(); i++) (*i).Print();

    /* Sort Warlords in the container */

    cout << endl << "Sorted Warlords" << endl;
    for (Iterator i = lords.begin(); i != lords.end(); i++) (*i).Print();

    Warlord w4; /* by default warlords have zero power and zero magic*/
    w4.Print();

    return 0;
}

```

Output:

Unsorted Warlords

Power: 1 Magic: 1

Power: 2 Magic: 2

Power: 2 Magic: 1

Sorted Warlords

Power: 2 Magic: 2

Power: 2 Magic: 1

Power: 1 Magic: 1

A warlord cannot be a non-magic wielder!