

Gyakorló feladatsor a ZH-ra való felkészüléshez

1. Írjunk függvényt, amely paraméterül kap egy C string-et, és azt úgy változtatja meg, hogy az utolsó karakterét a string elejére írja, a többi karakterét pedig ezután az eredeti sorrendben felsorolja.
2. Írjunk függvényt, amely paraméterként egy `int`-eket tartalmazó `vector` referenciát ill. egy `int` egészet kap, és a `for_each` függvény segítségével a `vector` összes elemét megszorozza a paraméterként kapott `int`-tel.
3. Írjunk egy template függvényt, amely paraméterként kap három értéket (ezeket konstans referenciaként adjuk át), amelyeknek típusa a template paraméter, nagyság szerinti sorrendbe rendezi őket, majd egy `vector`-ban visszaadja a rendezett hármaszt.
4. A következő kódban 3 darab hiba van, amik miatt nem fordul le. Keressük meg ezeket.

```
class C{
private:
    int a;
public:
    C() { a = 0; }
    C(int _a) :a(_a){}
    int getVal() { return a; }
    void setVal(int _a)const { a = _a; }
    bool equals(const C& other)const { return a == other.getVal(); }
};

int main() {
    C c1;
    C c2(2);
    c1.setVal(c2.a);
    return 0;
}
```

5. Mit ír ki a következő program a képernyőre és miért?

```
#include<iostream>

class Counter {
private:
    static int value;
    int val;
public:
    Counter(int _val) : val(_val) { value += val; }
    static int Value() { return value; }
    ~Counter() { value -= val; }
};

int Counter::value = 0;

int main() {
    Counter count(10);
    for (int i = 0; i < 5; ++i) {
        Counter tmp(i);
        std::cout << tmp.Value() << " ";
    }
    return 0;
}
```

6. Mit ír ki a képernyőre következő kód és miért?

```
#include <iostream>
using namespace std;

class base {
public:
    void fun_1() { cout << "base-1\n"; }
    virtual void fun_2() = 0;
    virtual void fun_3() { cout << "base-3\n"; }
};

class derived : public base {
public:
    void fun_1() { cout << "derived-1\n"; }
    void fun_2() { cout << "derived-2\n"; }
};

int main() {
    derived obj1;
    base* p = &obj1;

    p->fun_1();
    p->fun_2();
    p->fun_3();

    return 0;
}
```

7. Írjunk egy függvényt, amely két `double` értéket vár paraméterként, és visszaadja a hányadosukat, ha azonban a nevező 0, akkor `std::invalid_argument` kivételt dob. Írjunk egy `main` függvényt is, ami meghívja a fenti függvényt valamilyen paraméterrel, és elkapja a kivételt, ha a függvény dob, valamint kiírja a képernyőre a kivételben megadott hibaüzenetet.