

## 2. Zárthelyi dolgozat

1. A következő kód egy 3 dimenziós vektort implementál. Implementáljuk a következő konstruktorokat/metódusokat (feltételezhetjük, hogy a többi konstruktor/metódus helyesen van implementálva): (10 pont)

- `Vector()`;
- `Vector operator-(Vector& other)`;
- `float& operator[](int index)`;  
ez a vektor  $i$ . koordinátáját adja vissza. ( $0 \rightarrow x, 1 \rightarrow y, 2 \rightarrow z$ )

```
class Vector {
private:
    float x, y, z;
public:
    Vector();
    Vector(float x, float y);
    float length();
    Vector operator+(Vector& other);
    Vector operator-(Vector& other);
    Vector operator*(float scalar);
    float& operator[](int index);
};
```

2. A következő `Dolgozo` osztály egy alkalmazottat reprezentál. Feltételezhetjük, hogy helyesen van implementálva. A `havi_fizetes` adattag a fizetését reprezentálja. Az `eves_fizetes` kiszámolja, hogy egy év alatt mennyi fizetést kap összesen ( $12 * \text{havi\_fizetes}$ ). Írjunk egy `BonuszDolgozo` osztályt ami a `Dolgozo`-ból öröklődik és annyival tud többet, hogy neki van éves bónusza. (10 pont)

Ezeket implementáljuk a `BonuszDolgozo`-ban:

- konstruktor ami 1 int-et kap (a havi fizetését, ekkor a bónusz 0)
- konstruktor ami 2 int-et kap (a havi fizetését és a bónuszt)
- `eves_fizetes` (ebbe számítsuk bele az éves bónuszt is)

```
class Dolgozo {
protected:
    int havi_fizetes;
public:
    Dolgozo(int a);
    virtual int éves_fizetes();
};
```

Mutassunk példát arra (egy main függvényben) amikor számít, hogy a `Dolgozo` `eves_fizetes` metódusa virtual.

3. Mutassunk példát a `friend` kulcsszóra. Írjunk main függvényt, amiben használjuk ezt a függvényt. (10 pont)

4. Mit ír ki a következő kód?(10 pont)

```
class A {
public:
    int a[10];
    A(int x) {
        for(int i = 0; i < 10; i++) {
            a[i] = (i + 1) * x;
        }
    };
    int& select(int x, int y) {
        return a[x / y];
    }
    void func(int y) {
        for(int i = 0; i < 10; i++) {
            if(i % 2 == 0) {
                select(i, y) += 5;
            }
        }
    }
};
int main(void) {
    A obj = A(10);
    obj.func(2);
    for(int i = 0; i < 10; i++) {
        cout << obj.a[i] << endl;
    }
    return 0;
}
```

5. Adott a következő hiányos template függvény:(10 pont)

```
template<class T>
void print(T* a, int n) {

}
}
```

Egészítsük ki a függvényt, hogy tetszőleges típusú tömböknek tudja kiírni az elemeit szóközzel elválasztva. Írjunk egy main függvényt amiben használjuk 2 különböző típusú tömbbel.

Például ennek a tömbnek:

```
int a[] = {1,2,5};
```

Ilyen legyen a kiírása:

```
1 2 5
```