

## 10<sup>η</sup> Φροντιστηριακή Άσκηση στη C++: Πρότυπες Κλάσεις (Templates)

---

### Άσκηση 1: Πρότυπη Συνάρτηση Μέγιστου

#### Εκφώνηση:

Γράψτε μια πρότυπη συνάρτηση `maximum` που επιστρέφει το μεγαλύτερο από δύο δεδομένα οποιουδήποτε τύπου.

#### Λύση:

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

template <typename T>
T maximum(T a, T b) {
    return (a > b) ? a : b;
}

int main() {
    cout << "Μέγιστο μεταξύ 5 και 10: " << maximum(5, 10) << endl;
    cout << "Μέγιστο μεταξύ 3.5 και 2.1: " << maximum(3.5, 2.1) <<
endl;
    return 0;
}
```

## Άσκηση 2: Πρότυπη Κλάση Αποθήκευσης

### Εκφώνηση:

Γράψτε μια πρότυπη κλάση `Storage` που αποθηκεύει μία τιμή οποιουδήποτε τύπου και έχει μεθόδους για την ανάκτηση και την εμφάνιση της τιμής.

### Λύση:

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

template <typename T>
class Storage {
private:
    T value;
public:
    Storage(T v) : value(v) {}
    T getValue() { return value; }
    void display() { cout << "Αποθηκευμένη τιμή: " << value << endl; }
};

int main() {
    Storage<int> intStorage(42);
    Storage<string> stringStorage("Hello, Templates!");

    intStorage.display();
    stringStorage.display();
    return 0;
}
```

### Άσκηση 3: Πρότυπη Κλάση Δυναμικού Πίνακα

#### Εκφώνηση:

Γράψτε μια πρότυπη κλάση `Array` που αποθηκεύει ένα δυναμικό πίνακα και περιέχει μεθόδους για εισαγωγή και εμφάνιση των στοιχείων.

#### Λύση:

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

template <typename T>
class Array {
private:
    T* arr;
    int size;
public:
    Array(int s) : size(s) { arr = new T[size]; }
    void setElement(int index, T value) { if (index >= 0 && index <
size) arr[index] = value; }
    void display() {
        for (int i = 0; i < size; i++)
            cout << arr[i] << " ";
        cout << endl;
    }
    ~Array() { delete[] arr; }
};

int main() {
    Array<int> intArray(5);
    for (int i = 0; i < 5; i++)
        intArray.setElement(i, i * 2);

    intArray.display();
    return 0;
}
```

## Άσκηση 4: Πρότυπη Κλάση Ζεύγους (Pair)

### Εκφώνηση:

Γράψτε μια πρότυπη κλάση `Pair` που αποθηκεύει δύο τιμές διαφορετικών τύπων και περιέχει μεθόδους εμφάνισης.

### Λύση:

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

template <typename T1, typename T2>
class Pair {
private:
    T1 first;
    T2 second;
public:
    Pair(T1 f, T2 s) : first(f), second(s) {}
    void display() { cout << "Pair: (" << first << ", " << second <<
    ")" << endl; }
};

int main() {
    Pair<int, double> p1(10, 20.5);
    Pair<string, char> p2("C++", 'T');

    p1.display();
    p2.display();
    return 0;
}
```