

Άσκηση 1: Κλήση μεθόδου μέσω αντικειμένου

Εκφώνηση: Φτιάξε κλάση Person με όνομα και ηλικία και μέθοδο printInfo().
Κάλεσε τη μέθοδο από το αντικείμενο.

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

class Person {
public:
    string name;
    int age;

    void printInfo() {
        cout << name << " " << age << endl;
    }
};

int main() {
    Person p;
    p.name = "Maria";
    p.age = 25;

    p.printInfo(); // κλήση μεθόδου
}
```

Άσκηση 2: Συνάρτηση με αντικείμενο ως όρισμα

Εκφώνηση: Γράψε συνάρτηση showPerson(Person p) που εμφανίζει τα στοιχεία.

```
void showPerson(Person p) {
    p.printInfo();
}

int main() {
    Person p = {"Nikos", 30};
    showPerson(p);
}
```

Άσκηση 3: Πέρασμα με δείκτη

Εκφώνηση: Γράψε συνάρτηση που δέχεται δείκτη σε Person.

```
void showPerson(Person* p) {
    cout << p->name << " " << p->age << endl;
}

int main() {
    Person p = {"Anna", 22};
    showPerson(&p);
}
```

Άσκηση 4: Δείκτης σε αντικείμενο

Εκφώνηση: Δημιούργησε δείκτη σε αντικείμενο και κάλεσε μέθοδο.

```
int main() {
    Person p = {"John", 40};
    Person* ptr = &p;

    ptr->printInfo(); // μέσω δείκτη
}
```

Άσκηση 5: Δυναμική δέσμευση

Εκφώνηση: Δημιούργησε αντικείμενο με new και κάλεσε μέθοδο.

```
int main() {
    Person* p = new Person;

    p->name = "Chris";
    p->age = 35;

    p->printInfo();

    delete p;
}
```

Άσκηση 6: Πίνακας αντικειμένων

Εκφώνηση: Δημιούργησε πίνακα 3 αντικειμένων και εμφάνισε τα στοιχεία.

```
int main() {
    Person arr[3] = {
        {"A", 10},
        {"B", 20},
        {"C", 30}
    };

    for (int i = 0; i < 3; i++) {
        arr[i].printInfo();
    }
}
```

Άσκηση 7: Πίνακας δεικτών σε αντικείμενα

Εκφώνηση: Δημιούργησε πίνακα από δείκτες σε Person.

```
int main() {
    Person p1 = {"A", 10};
    Person p2 = {"B", 20};

    Person* arr[2];
    arr[0] = &p1;
    arr[1] = &p2;

    for (int i = 0; i < 2; i++) {
        arr[i]->printInfo();
    }
}
```

Άσκηση 8: Συνάρτηση που αλλάζει αντικείμενο

Εκφώνηση: Γράψε συνάρτηση που αυξάνει την ηλικία κατά 1 (με δείκτη).

```
void increaseAge(Person* p) {
    p->age++;
}

int main() {
    Person p = {"Maria", 20};
    increaseAge(&p);

    p.printInfo();
}
```

Άσκηση 9: Πίνακας δυναμικών αντικειμένων

Εκφώνηση: Δημιούργησε δυναμικά πίνακα αντικειμένων.

```
int main() {
    int n = 3;
    Person* arr = new Person[n];

    for (int i = 0; i < n; i++) {
        arr[i].name = "User";
        arr[i].age = i * 10;
    }

    for (int i = 0; i < n; i++) {
        arr[i].printInfo();
    }

    delete[] arr;
}
```

Άσκηση 10: Συνάρτηση με πίνακα αντικειμένων

Εκφώνηση: Γράψε συνάρτηση που δέχεται πίνακα Person και εμφανίζει τον μεγαλύτερο σε ηλικία.

```
void findOldest(Person arr[], int size) {
    int max = arr[0].age;
    int index = 0;

    for (int i = 1; i < size; i++) {
        if (arr[i].age > max) {
            max = arr[i].age;
            index = i;
        }
    }

    arr[index].printInfo();
}

int main() {
    Person arr[3] = {
        {"A", 20},
        {"B", 50},
        {"C", 30}
    };

    findOldest(arr, 3);
}
```