

# Βασικά Στοιχεία στη C++

Τι είναι η C++; Η C++ είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, γενικού σκοπού. Είναι ένας από τους πιο δημοφιλείς και ισχυρούς γλώσσες προγραμματισμού που χρησιμοποιούνται σε πολλές εφαρμογές όπως λογισμικό εφαρμογών, παιχνίδια, λογισμικό λειτουργικού συστήματος, και πολλά άλλα.

Βασική σύνταξη: Η C++ χρησιμοποιείται για την δημιουργία προγραμμάτων μέσω της γραφής πηγαίου κώδικα. Ένα απλό πρόγραμμα σε C++ ξεκινά με τη δήλωση της συνάρτησης `main()`. Η `main()` είναι η σημείο εκκίνησης του προγράμματος.

```
cpp

#include <iostream>

int main() {
    std::cout << "Hello, world!";
    return 0;
}
```

Σε αυτό το πρόγραμμα, η `#include <iostream>` χρησιμοποιείται για την εισαγωγή της βιβλιοθήκης εισόδου/εξόδου. Η συνάρτηση `main()` είναι όπου ξεκινά το πρόγραμμα. Η `std::cout` χρησιμοποιείται για την εκτύπωση κειμένου στην οθόνη.

Βασικές εντολές εισόδου/εξόδου: Στην C++, μπορούμε να χρησιμοποιήσουμε το `std::cin` για να λάβουμε είσοδο από τον χρήστη και το `std::cout` για να εμφανίσουμε έξοδο στην οθόνη.

```
cpp

#include <iostream>

int main() {
    int number;
    std::cout << "Enter a number: ";
    std::cin >> number;
    std::cout << "You entered: " << number;
    return 0;
}
```

Τύποι Δεδομένων: Στην C++, υπάρχουν διάφοροι τύποι δεδομένων που μπορούν να χρησιμοποιηθούν για να αποθηκεύσουν διαφορετικού είδους πληροφορίες. Ορισμένοι από τους κύριους τύπους δεδομένων περιλαμβάνουν:

1. Ακέραιοι Τύποι (Integer Types):

- `int`: Ακέραιοι αριθμοί.
- `short`: Σύντομοι ακέραιοι αριθμοί.
- `long`: Μεγάλοι ακέραιοι αριθμοί.
- `long long`: Πολύ μεγάλοι ακέραιοι αριθμοί.
- Π.χ.: `int x = 10;`

2. Δεκαδικοί Τύποι (Floating-Point Types):

- `float`: Μονής ακρίβειας δεκαδικοί αριθμοί.
- `double`: Διπλής ακρίβειας δεκαδικοί αριθμοί.
- `long double`: Πολύ υψηλής ακρίβειας δεκαδικοί αριθμοί.
- Π.χ.: `double y = 3.14;`

3. Χαρακτήρες (Character Types):

- `char`: Ένας χαρακτήρας.
- `char16_t`, `char32_t`: UTF-16 και UTF-32 χαρακτήρες αντίστοιχα.
- `wchar_t`: Ευρεία χαρακτήρας (wide character).
- Π.χ.: `char ch = 'A';`

4. Λογική (Boolean Type):

- `bool`: Αληθής ή ψευδής τιμή.
- Π.χ.: `bool flag = true;`

Πέραν από αυτούς τους βασικούς τύπους, μπορείς να ορίσεις τους δικούς σου τύπους δεδομένων μέσω των δομών (structs) ή των κλάσεων (classes), καθώς και να χρησιμοποιήσεις δείκτες (pointers) για να αναφέρεσαι σε μνήμη.

Οι τύποι δεδομένων επίσης μπορούν να συνδυαστούν με `qualifiers` όπως `const` για να οριστεί σταθερή τιμή ή `unsigned` για να δηλωθεί μη αρνητική τιμή.

```

#include <iostream>

int main() {
    // Ακέραιοι τύποι
    int num1 = 10;
    short num2 = 20;
    long num3 = 1234567890;
    long long num4 = 123456789012345;

    // Δεκαδικοί τύποι
    float floatNum = 3.14f;
    double doubleNum = 3.14159265359;
    long double longDoubleNum = 3.14159265358979323846;

    // Χαρακτήρες
    char character = 'A';

    // Λογική
    bool flag = true;

    std::cout << "Ακέραιοι:\n";
    std::cout << "int: " << num1 << "\n";
    std::cout << "short: " << num2 << "\n";
    std::cout << "long: " << num3 << "\n";
    std::cout << "long long: " << num4 << "\n";

```

```

    std::cout << "\nΔεκαδικοί:\n";
    std::cout << "float: " << floatNum << "\n";
    std::cout << "double: " << doubleNum << "\n";
    std::cout << "long double: " << longDoubleNum << "\n";

    std::cout << "\nΧαρακτήρας:\n";
    std::cout << "char: " << character << "\n";

    std::cout << "\nΛογική:\n";
    std::cout << "bool: " << std::boolalpha << flag << "\n";

    return 0;
}

```

Μεταβλητές: Οι μεταβλητές είναι απαραίτητες για την αποθήκευση δεδομένων σε ένα πρόγραμμα C++. Ας δούμε λίγα περισσότερα για τις μεταβλητές και τις δυνατότητές τους:

1. **Δήλωση Μεταβλητής:** Μια μεταβλητή πρέπει να δηλωθεί πριν από τη χρήση της. Η δήλωση μιας μεταβλητής περιλαμβάνει τον τύπο δεδομένων και το όνομα της μεταβλητής.

```
cpp
int x; // Δήλωση μιας μεταβλητής τύπου int με το όνομα x
```

2. **Ανάθεση Τιμής:** Μπορούμε να αναθέσουμε μια τιμή σε μια μεταβλητή χρησιμοποιώντας τον τελεστή ανάθεσης `=`.

```
cpp
x = 5; // Ανάθεση της τιμής 5 στην μεταβλητή x
```

3. **Αρχικοποίηση Μεταβλητής:** Μπορούμε να αρχικοποιήσουμε μια μεταβλητή κατά τη δήλωσή της.

```
cpp
int y = 10; // Δήλωση και αρχικοποίηση της μεταβλητής y με την τιμή 10
```

4. **Διεύθυνση Μνήμης (Δείκτες):** Οι δείκτες είναι μεταβλητές που δείχνουν σε μια διεύθυνση μνήμης. Μπορούμε να χρησιμοποιήσουμε τον τελεστή αναφοράς `&` για να ανακτήσουμε τη διεύθυνση μνήμης μιας μεταβλητής.

```
cpp
int z = 20;
int *ptr = &z; // Δήλωση ενός δείκτη που δείχνει στη μεταβλητή z
```

5. **Σταθερές:** Οι σταθερές είναι μεταβλητές οι οποίες δεν μπορούν να αλλάξουν τιμή κατά τη διάρκεια εκτέλεσης του προγράμματος. Η δήλωση σταθεράς γίνεται με τη χρήση του `const`.

```
cpp
const int MY_CONSTANT = 100; // Δήλωση μιας σταθεράς MY_CONSTANT με την τιμή 100
```

6. **Τελεστές Μεταβλητών:** Οι τελεστές χρησιμοποιούνται για να πραγματοποιούν διάφορες λειτουργίες σε μεταβλητές, όπως ο αριθμητικός υπολογισμός, οι λογικές πράξεις κλπ.

Οι τελεστές στη C++ είναι σύμβολα που χρησιμοποιούνται για να πραγματοποιήσουν διάφορες λειτουργίες σε μεταβλητές και τιμές. Αυτοί οι τελεστές διαίρονται σε διάφορες κατηγορίες, όπως αριθμητικοί τελεστές, λογικοί τελεστές, τελεστές σύγκρισης, τελεστές ανάθεσης και άλλοι. Ας δούμε μερικούς από τους βασικούς τελεστές και πώς λειτουργούν:

1. **Αριθμητικοί τελεστές:**

- `+`, `-`, `*`, `/`, `%`: Χρησιμοποιούνται για τις κοινές αριθμητικές πράξεις (πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση, υπόλοιπο).
- `++`, `--`: Χρησιμοποιούνται για την αύξηση ή μείωση μιας μεταβλητής κατά 1.

2. **Λογικοί τελεστές:**

- `&&`, `||`, `!`: Χρησιμοποιούνται για τη λογική σύγκριση και συνδυασμό λογικών τιμών (ΚΑΙ, Ή, Όχι).

3. **Τελεστές σύγκρισης:**

- `==`, `!=`, `<`, `>`, `<=`, `>=`: Χρησιμοποιούνται για τη σύγκριση δύο τιμών.

4. **Τελεστές ανάθεσης:**

- `=`, `+=`, `-=`, `*=`, `/=`, `%=`: Χρησιμοποιούνται για την ανάθεση τιμών σε μεταβλητές ή τη συνδυασμένη ανάθεση και λειτουργία.

5. **Άλλοι τελεστές:**

- `sizeof`: Χρησιμοποιείται για τον υπολογισμό του μεγέθους μιας μεταβλητής ή ενός τύπου δεδομένων.
- `?:`: Τριαδικός τελεστής (conditional operator), χρησιμοποιείται για την επιλογή μεταξύ δύο τιμών με βάση μια συνθήκη.
- `&`, `|`, `^`, `~`, `<<`, `>>`: Τελεστές bit-wise, χρησιμοποιούνται για λογικές πράξεις σε κάθε bit μιας μεταβλητής.

Αυτοί είναι μερικοί από τους βασικούς τελεστές στη C++. Υπάρχουν επίσης και άλλοι τελεστές που χρησιμοποιούνται για πιο εξειδικευμένες λειτουργίες.

```
#include <iostream>

int main() {
    int a = 10;
    int b = 5;

    // Αριθμητικοί τελεστές
    int sum = a + b;
    int difference = a - b;
    int product = a * b;
    int quotient = a / b;
    int remainder = a % b;

    std::cout << "Sum: " << sum << std::endl;
    std::cout << "Difference: " << difference << std::endl;
    std::cout << "Product: " << product << std::endl;
    std::cout << "Quotient: " << quotient << std::endl;
    std::cout << "Remainder: " << remainder << std::endl;

    // Λογικοί τελεστές
    bool isGreater = (a > b);
    bool isEqual = (a == b);
    bool logicalAnd = (isGreater && isEqual);
    bool logicalOr = (isGreater || isEqual);
    bool logicalNot = !isEqual;
```

```

std::cout << "isGreater: " << std::boolalpha << isGreater << std::endl;
std::cout << "isEqual: " << isEqual << std::endl;
std::cout << "logicalAnd: " << logicalAnd << std::endl;
std::cout << "logicalOr: " << logicalOr << std::endl;
std::cout << "logicalNot: " << logicalNot << std::endl;

// Τελεστές ανάθεσης
int x = 10;
x += 5; // ισοδύναμο με x = x + 5;
std::cout << "x after +=: " << x << std::endl;

// Τριαδικός τελεστής
int y = (a > b) ? a : b;
std::cout << "y (maximum of a and b): " << y << std::endl;

return 0;
}

```

Σε αυτό το παράδειγμα, χρησιμοποιούμε αριθμητικούς τελεστές για να πραγματοποιήσουμε πράξεις μεταξύ των μεταβλητών  $a$  και  $b$ . Επίσης, χρησιμοποιούμε λογικούς τελεστές για να ελέγξουμε τις σχέσεις μεταξύ τους. Η εντολή `int y = (a > b) ? a : b;` χρησιμοποιεί τον τριαδικό τελεστή `?:`, γνωστό και ως τριαδική συνθήκη ή conditional operator.

Αυτή η εντολή λειτουργεί ως εξής:

1. Εκτιμάται η συνθήκη  $(a > b)$ . Αν αυτή η συνθήκη είναι αληθής, τότε η τιμή του  $y$  θα είναι η τιμή της μεταβλητής  $a$ .
2. Αν η συνθήκη είναι ψευδής, τότε η τιμή του  $y$  θα είναι η τιμή της μεταβλητής  $b$ .

Με άλλα λόγια, αυτός ο τριαδικός τελεστής επιλέγει μεταξύ δύο τιμών ανάλογα με την αλήθεια ή τη ψευδαιμία μιας συνθήκης.

Σε αυτήν την περίπτωση, εάν η τιμή της μεταβλητής  $a$  είναι μεγαλύτερη από την τιμή της μεταβλητής  $b$ , τότε η τιμή της μεταβλητής  $y$  θα είναι η τιμή της μεταβλητής  $a$ . Διαφορετικά, η τιμή της  $y$  θα είναι η τιμή της  $b$ .