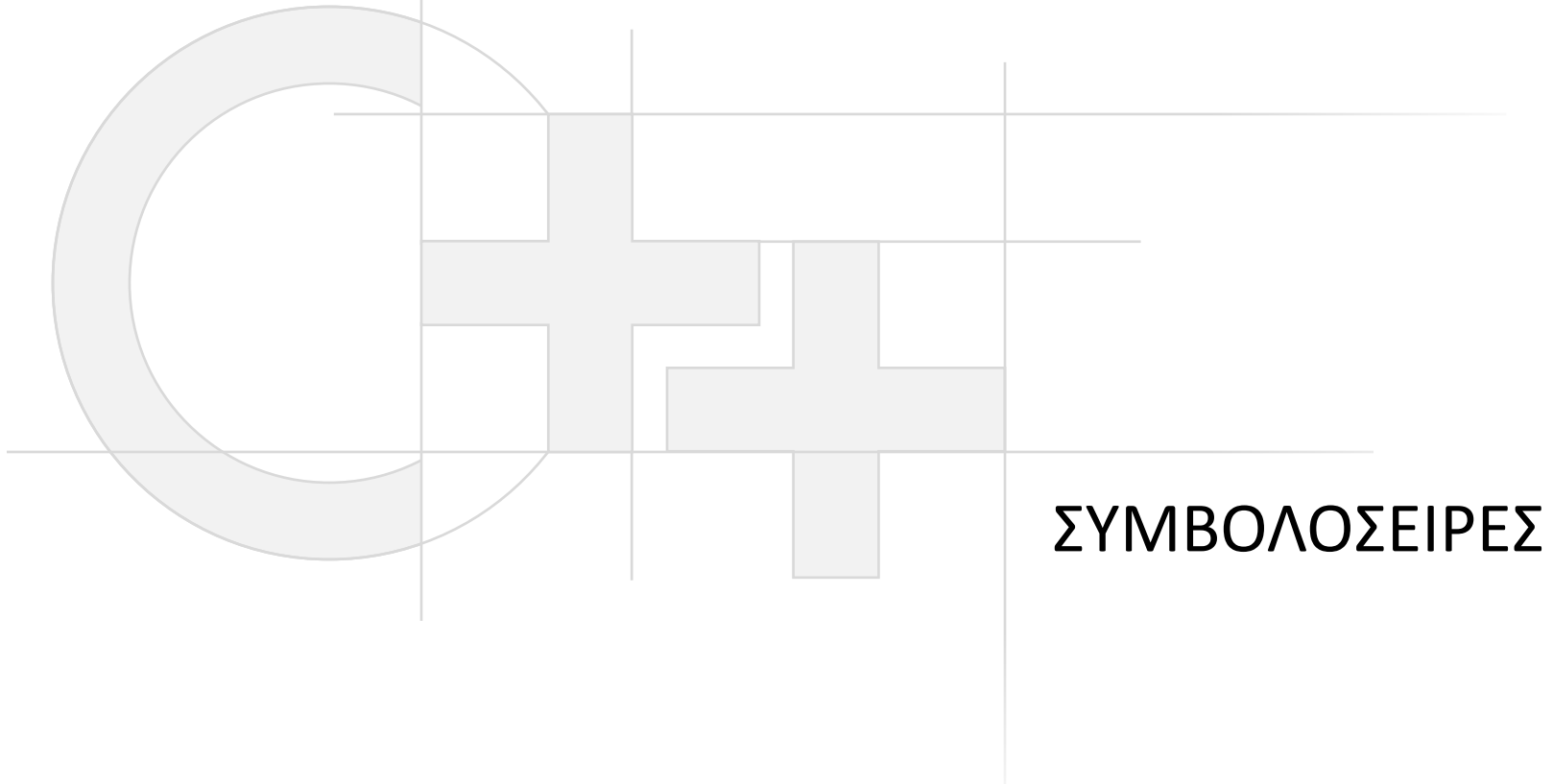


Προγραμματισμός Υπολογιστών με C++



ΣΥΜΒΟΛΟΣΕΙΡΕΣ

Περιεχόμενο Παρουσίασης

- Περιγραφή:
 - Ο τύπος string
 - Μετατροπή από και προς τον τύπο string
 - Βασικές μέθοδοι
 - Χρήση Ελληνικών
 - Συναρτήσεις C εκτύπωσης και ανάγνωσης
- Τελευταία ενημέρωση: Οκτώβριος 2020

Συμβολοσειρές

- Είδαμε ότι τα null-terminated strings (πίνακες χαρακτήρων) μπορούν να χρησιμοποιηθούν ως strings
- Στη C++ υπάρχει ειδική κλάση, η string που ενσωματώνει μια αντικειμενοστρεφή υλοποίηση των συμβολοσειρών και παρέχει υποστήριξη για σύνθετες αναπαραστάσεις χαρακτήρων, όπως Unicode, wide characters κλπ.
- Σημείωση: Εξακολουθούμε να δουλεύουμε με πίνακες χαρακτήρων όταν:
 - Θέλουμε συμβατότητα με C (γίνεται και μετατροπή μεταξύ των 2 τύπων)
 - Επιθυμούμε μεγαλύτερη ταχύτητα (απλούστερη αναπαράσταση)

Η Κλάση string

```
#include <string>           // Εδώ δηλώνεται το string
#include <iostream>
using namespace std;       // Εδώ ανήκει το string (std::string)

int main() {
    string s1, s2;         // Δε χρειάζεται να καθορίσουμε τα μεγέθη.
    cin >> s1;            // Ο χώρος του s1 προσαρμόζεται αυτόματα.
    s2 = "copy of " + s1;  // Πρόσθεση συμβολοσειρών. Μετατροπή
                          // του πίνακα χαρακτήρων σε string.
    s1 = "new string";     // Νέα τιμή του s1.
    cout << s1 << endl << s2 << endl;
}
```

Πληκτρολογή "test". Τυπώνει:

```
new string
copy of test
```

std::string : Παρατήρηση

- Η κλάση `string` είναι μέρος της βασικής βιβλιοθήκης «standard template library» (STL) η οποία είναι σπασμένη σε διάφορα αρχεία (π.χ. `iostream`, `string` κλπ)
- Η βιβλιοθήκη αυτή εντάσσει όλες τις κλάσεις της, τους τύπους της και τις συναρτήσεις της σε έναν κοινό χώρο ονομάτων (`namespace`), τον `std`
- Για να δηλώσω ένα `string` κανονικά ενσωματώνω και το `namespace` στο όνομα της κλάσης: `std::string`
- Αν δηλώσω ότι «χρησιμοποιώ» αυτό το `namespace`, μπορώ να το παραλείψω από τη δήλωση:

```
using namespace std;
```

```
string var = "my string";
```

Μετατροπή από και προς Πίνακες Χαρακτήρων

- Από πίνακα χαρακτήρων σε string:

```
char buffer[20] = "try me";  
std::string str = buffer;
```

- Από string σε πίνακα χαρακτήρων:

```
str.c_str();
```

- Πρόσβαση στον εσωτερικό buffer (χαρακτήρων):

```
str.data(); // Δεν είναι αντίγραφο και  
           // δε μπορούμε να το αλλάξουμε (const)
```

Μετρικές πάνω σε strings

- Μέγεθος:

`string.size()` και `string.length()` (ισοδύναμα)

- Άδεια συμβολοσειρά:

`string.empty()`

- Ανατρέξτε στον οδηγό αναφοράς για τον πλήρη κατάλογο των μεθόδων του string:

<http://www.cplusplus.com/reference/string/string/>

Βασικές Πράξεις με strings ⁽¹⁾

- Σύνθεση συμβολοσειρών:

```
string str1 = "try me ";  
string str2 = "now";  
string str3 = str1 + str2;    // str3: "try me now"  
str1.append(str2);           // str1: "try me now"  
str1 += " again";           // (append) "try me now again"
```

- Πρόσβαση σε χαρακτήρες:

```
cout << str3[2] << endl;    // διάβασμα του 3ου χαρ.: 'y'  
str3[4] = 'i';              // αντικατάσταση του 5ου χαρ.  
str3[5] = 't';              // αντικατάσταση του 6ου χαρ.  
cout << str3 << endl;     // "try it now"
```


Βασικές Πράξεις με strings (2)

- Σύγκριση συμβολοσειρών:

```
string str1 = "Android";  
string str2 = "Androids";  
  
bool _2gt1 = str1 < str2;    // _2gt1 == true  
  
str1 = "Android4.2";  
str2 = "Android3.2";  
  
_2gt1 = str1 < str2;    // _2gt1 == false  
  
str1 = "Andro";  
str2 = "Andras";  
  
_2gt1 = str1 < str2;    // _2gt1 == false
```

Ο έλεγχος γίνεται λεξικογραφικά. Το αποτέλεσμα ισοδυναμεί με τη σύγκριση των πρώτων από αριστερά ανόμοιων χαρακτήρων

Βασικές Πράξεις με strings (3)

- Εύρεση sub-string:

```
string str1 = "I 2 am a healthy 16 camel";  
string str2 = "am";  
size_t pos = str1.find(str2);    // pos == 2  
pos = str1.find(str2, pos+1);    // pos == 16
```

Μετατροπή Αριθμών σε string: to_string

```
string to_string (int val);  
string to_string (long val);  
string to_string (long long val);  
string to_string (unsigned val);  
string to_string (unsigned long val);  
string to_string (unsigned long long val);  
string to_string (float val);  
string to_string (double val);  
string to_string (long double val);
```

Μετατροπή string σε Αριθμό (sto[i/l/ul/ll/f/d/ld])

```
int stoi (const string& str, size_t* idx = 0, int base = 10);
```

```
long stol (const string& str, size_t* idx = 0, int base = 10);
```

```
unsigned long stoul (const string& str, size_t* idx = 0, int  
base = 10);
```

```
long long stoll (const string& str, size_t* idx = 0, int base =  
10);
```

```
unsigned long long stoull (const string& str, size_t* idx = 0,  
int base = 10);
```

```
float stof (const string& str, size_t* idx = 0);
```

```
double stod (const string& str, size_t* idx = 0);
```

```
long double stold (const string& str, size_t* idx = 0);
```

Συναρτήσεις C για Εκτύπωση/Διάβασμα strings

- Η κλάση `string` είναι της C++ (η C δεν είναι αντικειμενοστρεφής)
- Όμως, η C παρέχει ένα μεγάλο εύρος συναρτήσεων για πίνακες χαρακτήρων
- Πολύ βασικές συναρτήσεις για μορφοποιημένη είσοδο και έξοδο συμβολοσειρών:
 - `printf, sprintf`
 - `scanf, sscanf`

Η συνάρτηση printf

- Τυπώνει παραμετρικό μορφοποιημένο κείμενο
- Παίρνει ως είσοδο μια συμβολοσειρά μορφοποίησης και ένα πλήθος παραμέτρων που «κουμπώνουν» σε κατάλληλες θέσεις μέσα σε αυτή
- Παράδειγμα:

```
int hits = 33;  
long total = 1000;  
printf( "We scored %d out of %ld points.\nThis is %.2f%%.\n",  
        hits, total, 100.f*hits/(float)total );
```

Τυπώνει:

```
We scored 33 out of 1000 points.
```

```
This is 3.30%.
```

Ανάλυση της printf

```
printf( "We scored %d out of %ld points.\nThis is %.2f%%.\n",  
       hits, total, 100.f*hits/(float)total );
```

Θέση 1 Θέση 2 Θέση 3

Παράμετρος 1 Παράμετρος 2 Παράμετρος 3

- % : δηλώνει ότι ακολουθεί εκτύπωση παραμέτρου
 - %d : εκτυπώνει δεκαδικό αριθμό (**d**ecimal)
 - %ld: εκτυπώνει «μεγάλο» δεκαδικό (**l**ong **d**ecimal)
 - %f: εκτυπώνει float (και double)
 - ... Πλήρης λίστα: <http://www.cplusplus.com/reference/cstdio/printf/>
- %%: δηλώνει την εμφάνιση του χαρακτήρα %
- %.2f: Θέλουμε να εμφανίζει 2 δεκαδικά ψηφία

Η Συνάρτηση scanf

- Διαβάζει μορφοποιημένα δεδομένα από το ρεύμα εισόδου και περνάει σε μεταβλητές τα ορίσματα που εμφανίζονται στις θέσεις που υποδεικνύουμε
- Είναι ουσιαστικά η αντίθετη της printf

```
char name[80];  
  
int age;  
  
int read = scanf ("name: %s, %d ", name, &age);
```
- Αν αποτύχει η ανάγνωση γιατί η συμβολοσειρά εισόδου δεν ταιριάζει στο μορφότυπο που δώσαμε, επιστρέφει αμέσως
- Η τιμή επιστροφής είναι ο αριθμός των παραμέτρων που διαβάστηκαν ορθά

Η scanf και η sprintf

- Χρήσιμες συναρτήσεις για μορφοποιημένη έξοδο και είσοδο σε/από πίνακες χαρακτήρων
- Χρήσιμες για εύκολη επεξεργασία συμβολοσειρών (π.χ. parsing γραμμών κειμένου από αρχεία)
- Έχουν ένα παραπάνω όρισμα (το πρώτο), τον πίνακα χαρακτήρων εισόδου ή εξόδου (δείτε τον οδηγό αναφοράς για περισσότερες πληροφορίες)