



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων Τοπογράφων Μηχανικών

Εισαγωγή στην πληροφορική

Βασίλειος Βεσκούκης

Δρ. Ηλεκτρολόγος Μηχανικός &
Μηχανικός Υπολογιστών ΕΜΠ
v.vescoukis@cs.ntua.gr

Ρωμύλος Κορακίτης

Αστροφυσικός
Αναπλ. Καθηγητής ΕΜΠ
romylos@survey.ntua.gr

Είσοδος - έξοδος δεδομένων

Μορφοποίηση εξόδου στη C++

Παραδείγματα

Ρεύματα εισόδου / εξόδου (I / O streams)

Ρεύμα εισόδου (input stream) : μια ακολουθία χαρακτήρων **από** μια συσκευή εισόδου (πχ. πληκτρολόγιο, αρχείο κλπ) **προς** το σύστημα. (τελεστής εισαγωγής **>>**)

Ρεύμα εξόδου (output stream) : μια ακολουθία χαρακτήρων **από** το σύστημα **προς** μια συσκευή εξόδου (πχ. οθόνη, αρχείο κλπ). (τελεστής εξαγωγής **<<**)

Για να χρησιμοποιηθούν τα ρεύματα πρέπει να δηλωθεί η βιβλιοθήκη `iostream`

Γνωρίζουμε ήδη

- το καθιερωμένο ρεύμα εισόδου **cin** (**console input** = πληκτρολόγιο)
- το καθιερωμένο ρεύμα εξόδου **cout** (**console output** = οθόνη)

αυτά δηλώνονται στο χώρο ονομάτων `standard`

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

Είσοδος δεδομένων

- Συνήθης χρήση: `cin>>[όνομα μεταβλητής] ;`
- Επιτρέπεται και η χρήση: `cin>>[var1]>>[var2]>>...>>[varN] ;`

Παράδειγμα

```
int a, b, c;  
char s;
```

...

```
cin>>a; // διαβάζεται από το πληκτρολόγιο η τιμή του (ακεραίου) a  
cin>>s; // το ίδιο για την τιμή του (χαρακτήρα) s
```

Όμως **δεν μπορούμε να ανακατέψουμε είσοδο και έξοδο!**

Δηλαδή: **Το cin δεν μπορεί να εμφανίζει μηνύματα στην οθόνη!**

```
cin>>"Enter the value of a :">>a; ΛΑΘΟΣ
```

Γι αυτό είναι καλό να χρησιμοποιείται και η εμφάνιση στην οθόνη κατάλληλων μηνυμάτων προς τον χρήστη με χρήση του ρεύματος εξόδου. Παράδειγμα:

```
int n;  
cout<<"Enter value of integer variable n: ";  
cin>>n;
```

Έξοδος δεδομένων

Έξοδος στην οθόνη

- Μηνυμάτων που δίνονται κυριολεκτικά (string literals)
- Τιμών μεταβλητών μνήμης
- Συνδυασμού των παραπάνω

```
cout<<"This is a message";
```

```
// τυπώνει ακριβώς τα περιεχόμενα των εισαγωγικών
```

```
cout<<k;
```

```
// τυπώνει την τιμή της μεταβλητής k
```

```
cout<<"k";
```

```
// τυπώνει το γράμμα k
```

```
cout<<"k="<<k;
```

```
// (έστω ότι η k έχει την τιμή 10)
```

```
// τυπώνει k=10
```

Έξοδος δεδομένων

Ειδικοί χειριστές ροής

- `\n` (μέσα σε εισαγωγικά): κατεβαίνει στην επόμενη γραμμή
- `endl`: κατεβαίνει στην επόμενη γραμμή

Παράδειγμα

```
cout<<"\nThis is a new day!\n";  
cout<<endl<<"This is a new day!"<<endl;
```

Συνήθης χρήση

```
cout<<[μήνυμα που θέλουμε να εμφανίζεται πριν από την είσοδο] ;  
cin>> [μεταβλητή την τιμή της οποίας διαβάζουμε];  
Η είσοδος γίνεται μέσω του cin>>
```

Παράδειγμα

```
char s;  
...  
cout<<"Enter the value of s:";  
cin>>s;
```

Έλεγχος τιμών εισόδου

```
do {  
    /*εμφάνιση μηνύματος */  
    /*ανάγνωση τιμών*/  
} while [συνθήκη ΜΗ ΕΓΚΥΡΩΝ τιμών];
```

Παραδείγματα

- do {
 cout<<endl<<"Enter the value of a:";
 cin>>a;
} while (!(a>=0) && (a<=9)) // αποδέχεται τιμές 0-9
- do {
 cout<<endl<<"Enter the value of a:";
 cin>>a;
 cout<<endl<<"Enter the value of b:";
 cin>>b;
} while (a*b==0); // αποδέχεται τιμές a!=0 και b!=0

Είσοδος δεδομένων - 2

Παράδειγμα σύνθετης εισόδου δεδομένων:

```
char ch1, ch2;  
int num;  
cin>>ch1>>ch2>>num;
```

Ο χρήστης πληκτρολογεί : **A 25** Τι θα συμβεί;

Είσοδος δεδομένων - 2

Παράδειγμα σύνθετης εισόδου δεδομένων:

```
char ch1, ch2;  
int num;  
cin>>ch1>>ch2>>num;
```

Ο χρήστης πληκτρολογεί : **A 25** Τι θα συμβεί;

Όταν εκτελεστεί η εντολή εισόδου, οι μεταβλητές θα έχουν τιμές:

```
ch1 <-- A  
ch2 <-- 2  
num <-- 5
```

Αιτία: ο τελεστής εισαγωγής >> αγνοεί τα κενά

Είσοδος δεδομένων - 2

Παράδειγμα σύνθετης εισόδου δεδομένων:

```
char ch1, ch2;  
int num;  
cin>>ch1>>ch2>>num;
```

Ο χρήστης πληκτρολογεί : **A 25** Τι θα συμβεί;

Όταν εκτελεστεί η εντολή εισόδου, οι μεταβλητές θα έχουν τιμές:

```
ch1 <-- A  
ch2 <-- 2  
num <-- 5
```

Αιτία: ο τελεστής εισαγωγής >> αγνοεί τα κενά

Πως μπορούμε να δώσουμε τις ακόλουθες τιμές για τις μεταβλητές:

```
ch1 <-- A  
ch2 <-- (κενό)  
num <-- 25
```

Είσοδος δεδομένων - 2

Παράδειγμα σύνθετης εισόδου δεδομένων:

Χρησιμοποιούμε την μέθοδο `get` του ρεύματος εισόδου, με γενική μορφή:
`cin.get(varChar);`

```
char ch1, ch2;  
int num;  
cin.get(ch1);  
cin.get(ch2);  
cin>>num;
```

Όταν εκτελεστεί η εντολή εισόδου, οι μεταβλητές θα έχουν τιμές:

```
ch1 <-- A  
ch2 <-- (κενό)  
num <-- 25
```

Παράδειγμα: αρχείο `cs2005_lec09_ex1.cpp`

Μορφοποίηση δεδομένων εξόδου

Η γενική μορφή χρήσης του ρεύματος εξόδου είναι:

```
cout<<παράσταση ή διαμορφωτής<<παράσταση ή διαμορφωτής...;
```

όπου παράσταση είναι αριθμητική ή αλφαριθμητική σταθερά ή όνομα μεταβλητής

Ο απλούστερος διαμορφωτής (ή χειριστής ροής - manipulator) είναι ο **endl**.

Χρήσιμοι είναι ακόμη οι χειριστές μέσα σε κείμενο:

- ο χειριστής tab **"\t"**
- ο χειριστής τέλους γραμμής **endl** **"\n"**

Άλλοι χρήσιμοι διαμορφωτές είναι:

- **fixed** : εμφάνιση αριθμών με σταθερή υποδιαστολή
- **scientific** : εμφάνιση αριθμών με εκθετική μορφή και κινητή υποδιαστολή
- **setprecision(d)** : εμφάνιση **d** δεκαδικών ψηφίων μετά την υποδιαστολή
- **setw(n)** : χρήση πεδίου αριθμητικής εξόδου συνολικού εύρους **n** χαρακτήρων

Οι διαμορφωτές αυτοί απαιτούν την χρήση της βιβλιοθήκης **iomanip**

```
#include <iomanip>
```

Παράδειγμα: αρχείο `cs2005_lec09_ex2.cpp`