



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Οντοκεντρικός Προγραμματισμός

Ενότητα 9: C++ ΕΙΣΟΔΟΣ - ΕΞΟΔΟΣ / ΑΛΦΑΡΙΘΜΗΤΙΚΑ / ΑΡΧΕΙΑ

## Κανάλια Εισόδου - Εξόδου

Ιωάννης Χατζηλυγερούδης, Χρήστος Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Είσοδος / Έξοδος (Streams)

# Βιβλιοθήκη `iostream`

---

- Βιβλιοθήκη `iostream`
  - Περιέχει εκατοντάδες βιβλιοθήκες για δυνατότητες Εισόδου/Εξόδου
  - `<iostream.h>`
    - Κύρια Είσοδος (`cin`)
    - Κύρια Έξοδος (`cout`)
    - Unbuffered error (`cerr`)
    - Buffered error (`clog`)
  - `<iomanip.h>`
    - Μορφοποιημένη Είσοδος/Έξοδος με stream manipulators
  - `<fstream.h>`
    - Λειτουργίες επεξεργασίας αρχείων



# Βιβλιοθήκη `iostream`

- Η βιβλιοθήκη `iostream` διαθέτει templates:
  - `basic_istream` (είσοδος από κανάλι - stream)
  - `basic_ostream` (έξοδος σε κανάλι - stream)
  - `basic_iostream` (είσοδος και έξοδος απο/σε κανάλι)
- Η `typedef` ορίζει ψευδώνυμα για τύπους δεδομένων
  - `typedef Card *CardPtr;`
    - `CardPtr` συνώνυμο με `Card *`
  - `typedefs istream, ostream, iostream`
    - Επιτρέπουν είσοδο/έξοδο χαρακτήρων



# Κύρια Κανάλια

- Τελεστές `<<` και `>>`
  - Τελεστές εισόδου και εξόδου από κανάλι (stream)
- **cin**
  - Είναι αντικείμενο **istream**
  - Συνδέεται με την κύρια είσοδο (συνήθως το πληκτρολόγιο)
  - **cin >> var;**
    - Ο Compiler αποφασίζει τον τύπο της μεταβλητής var
    - Καλεί την κατάλληλη υπερφορτωμένη συνάρτηση
- **cout**
  - Είναι αντικείμενο **ostream** object
  - Συνδέεται με την κύρια έξοδο (συνήθως η οθόνη)
  - **cout << var;**
    - Όμοια
- **cerr, clog**
  - Αντικείμενα **ostream**
  - Σύνδεση με το κυρίως κανάλι σφαλμάτων
  - Το **cerr** βγάζει την έξοδο απευθείας
  - Το **clog** κρατάει την έξοδο σε buffer



# Επεξεργασία Αρχείων

---

- Η επεξεργασία αρχείων στην C++ είναι παρόμοια:
  - Class templates
    - `basic_ifstream` (είσοδος από αρχείο)
    - `basic_ofstream` (έξοδος σε αρχείο)
    - `basic_fstream` (είσοδος και έξοδος απο/σε αρχείο)
  - Επιτρέπουν είσοδο/έξοδο χαρακτήρων:
    - `typedefs`
    - `ifstream`
    - `ofstream`
    - `fstream`



# Είσοδος / Έξοδος

## ■ Έξοδος

- Χρήση **ostream**
- Χρήση του τελεστή << για τους βασικούς τύπους
- Χαρακτήρες (χρήση της συνάρτησης **put**)
- Ακέρατοι (δεκαδικοί, οκταδικοί, δεκαεξαδικοί)
- Κινητής Υποδιαστολής
  - Ακρίβεια, δεκαδικά ψηφία, επιστημονική σήμανση
- Justified, padded data
- Έλεγχος κεφαλαίων/πεζών

## ■ Είσοδος

- Χρήση **istream**
- Χρήση του τελεστή >>
- Επιστρέφει 0 όταν συναντήσει EOF
  - Διαφορετικά επιστρέφει αναφορά στο ρεύμα εισόδου
  - **cin >> var**
- Αν προκύψει σφάλμα ενημερώνονται τα State bits



# Είσοδος Χαρακτήρα

---

- **get** συνάρτηση
  - **cin.get()**
  - Επιστρέφει έναν χαρακτήρα ή EOF
- End-of-file (EOF)
  - σηματοδοτεί το τέλος της εισόδου
  - Ανάλογα το σύστημα: *ctrl-z* ή *ctrl-d*
  - **cin.eof()**
    - Επιστρέφει **1 (true)** αν έχει συμβεί το EOF





# Παράδειγμα

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;
int main() {
    int character;
    cout << "EOF" << cin.eof() << endl
    while ( ( character = cin.get() ) != EOF )
        cout.put( character );
    cout << "\nEOF in this system is: " << character << endl;
    cout << "EOF" << cin.eof() << endl return 0; }
```

```
cin.EOF:0
test sentence
test sentence
^Z
Last Charachter entered (EOF): -1
cin.EOF::1
```



# Είσοδος χαρακτήρα/ων

- **get (charRef)**
  - Ως όρισμα παίρνει χαρακτήρα με αναφορά
  - Ανακτά έναν χαρακτήρα και τον αποθηκεύει στην **charRef**
    - Returns reference to **istream**
    - Επιστρέφει -1, αν διαβάσει EOF
- **get (charArray, size, delimiter)**
  - Διαβάζει έως **size-1** χαρακτήρες, μέχρι να σηναντήσει τον χαρακτήρα **delimiter**
    - Αν δεν δοθεί **delimiter** θεωρείται ο **' \n '**
    - Ο **delimiter** **παραμένει** στο κανάλι εισόδου
      - Μπορεί να αφαιρεθεί με **cin.get ()** or **cin.ignore ()**
  - Τερματίζει το **charArray** με τον χαρακτήρα τερματισμού **' \0 '**
- **getline (array, size, delimiter)**
  - Όμοια με την παραπάνω **get** με τα ίδια ορίσματα
  - **Αφαιρεί** τον χαρακτήρα **delimiter** από το κανάλι εισόδου



# Παράδειγμα

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;
int main(){
    char buf1[ 60 ];
    char buf2[ 60 ];
    cout<<"Your Sentence:";
    cin >> buf1;
    cout <<"Buf1:" buf1 <<endl;
    cin.get( buf2, 60 );
    cout <<"Buf2:" buf2 ;
}
```

Your Sentence:Soon comes the cold, and the night that never ends.  
Buf1:Soon  
Buf2: comes the cold, and the night that never ends.



# Είσοδος χαρακτήρα/ων

---

- **ignore ()**
  - Αφαιρεί χαρακτήρες από το κανάλι εισόδου (default 1)
  - Σταματάει όταν συναντήσει delimiter (Default **EOF**)
- **putback ()**
  - Τοποθετεί πίσω στο κανάλι τον χαρακτήρα που ανακτήθηκε με την **get ()**
- **peek ()**
  - Επιστρέφει τον επόμενο χαρακτήρα του καναλιού, χωρίς να τον αφαιρέσει



# Error Bits

---

- Οι τελεστές << και >> είναι υπερφορτωμένοι ώστε να αποδέχονται διάφορους τύπους
- Αν προκύψουν μη αναμενόμενου τύπου δεδομένα τότε ενημερώνονται κατάλληλα **Error Bits** τα οποία μπορούμε να ελέγξουμε για να διαπιστώσουμε αν απέτυχε η εντολή.



# Είσοδος / έξοδος Bytes

---

- **read (istream)**

- Είσοδος **bytes** σε πίνακα χαρακτήρων
- Αν δεν ανακτηθούν αρκετοί χαρακτήρες ενημερώνεται το **failbit**
- Η **gcount ()** επιστρέφει πόσοι χαρακτήρες ανακτήθηκαν στην τελευταία εντολή εισόδου

- **write (ostream)**

- Έξοδος **bytes** από πίνακα χαρακτήρων μέχρι να συναντήσει **null**



# String Manipulators

---

- Οι παραπονητές καναλιού επιτελούν λειτουργίες μορφοποίησης όπως:
  - Πλάτος (σε χαρακτήρες)
  - Ακρίβεια αριθμών (Precision)
  - Σημαίες Μορφοποίησης
  - Γέμισμα χαρακτήρων
  - Flushing καναλιού
  - Είσοδος νέας γραμμής σε κανάλι εξόδου
  - Αγνόηση κενών από κανάλι εισόδου



# Παραποίηση Ακεραίων

- Κανονικά η βάση των ακεραίων θεωρείται το 10
  - Μπορούμε να αλλάξουμε βάση με χρήση των:
    - **hex** (βάση το 16)
    - **oct** (βάση το 8)
    - **dec** (επαναφέρει ως βάση το 10)
  - **cout << hex << myInt**
  - **setbase(newBase)** συνάρτηση
    - Δύνατες τιμές: 8, 10, or 16
  - Η παραποίηση ισχύει μέχρι να την αλλάξουμε ρητά.

```
int number;  
cin >> number;  
cout << number << " in hexadecimal is: " << hex  
    << number << endl;
```





# Παραποίηση Δεκαδικών

---

- Παραποίηση της ακρίβειας αριθμών κινητής υποδιαστολής:
  - Αριθμός ψηφίων μετά την υποδιαστολή
  - **setprecision**
    - Ως όρισμα παίρνει πόσα δεκαδικά ψηφία θα εμφανιστούν
    - **cout << setprecision(5)**
  - **precision** συνάρτηση
    - **cout.precision(newPrecision)**
  - Η παραποίηση ισχύει μέχρι να την αλλάξουμε ρητά.



# Παραποίηση Δεκαδικών

- **width** συνάρτηση (κλάσης `ios_base`)
  - `cin.width(5)`
  - Καθορίζει το πλάτος (σε χαρακτήρες)
    - Αριθμός χαρακτήρων στην έξοδο
    - Μέγιστος αριθμός στην είσοδο
  - Εναλλακτικά μπορεί να χρησιμοποιηθεί ο `setw` manipulator ( `<< setw` )
  - Όταν διαβάζει πίνακα χαρακτήρων διαβάζει έναν λιγότερο για να τοποθετήσει στο τέλος το `null`
- **showpoint**
  - Αναγκάζει δεκαδικούς να εκτυπωθούν με συγκεκριμένο αριθμό χαρακτήρων προσθέτοντας μηδενικά στο τέλος (πχ 15.400000 αντί για 15.4)
  - Επαναφορά με την `noshowpoint`



# Στοίχιση / Πρόσημο

---

- **left** manipulator
  - Στοίχιση αριστερά με γέμισμα των θέσεων δεξιά
- **right** manipulator
  - Στοίχιση δεξιά με γέμισμα των θέσεων αριστερά
- **internal**
  - Το πρόσημο στοιχίζεται αριστερά, η τιμή δεξιά
$$+ \quad 123$$
- **showpos**
  - υποχρεώνει την εκτύπωση του προσήμου
  - Επαναφορά με **noshowpos**



# Χαρακτήρας γεμίσματος

---

- Ορισμός του χαρακτήρα που θα χρησιμοποιηθεί όταν χρειαστεί να καλυφθούν θέσεις
  - **fill**                    συνάρτηση  

```
cout.fill('*')
```
  - **setfill**                manipulator  

```
<< setfill( '^' )
```



# Κεφαλαία / Πεζά

---

- **uppercase** manipulator
  - Μετατρέπει χαρακτήρες σε κεφαλαία
  - lowercase η προκαθορισμένη συμπεριφορά
  - Επαναφορά με **nouppercase**



# Έλεγχος Σφαλμάτων

---

- **rdstate ( )** συνάρτηση
  - Επιστρέφει την κατάσταση σφαλμάτων του καναλιού
  - Ελέγχει για **goodbit**, **badbit**, etc.
  - Συνιστάται η χρήση των: **good ( )**, **bad ( )**
- **clear ( )** συνάρτηση
  - Προκαθορισμένη τιμή ορίσματος: **goodbit**
  - Θέτει την κατάσταση σε "good", ώστε να συνεχίσει η είσοδος/έξοδος
  - Can pass other values
    - **cin.clear( ios::failbit )**  
Αλλάζει τιμή στο **failbit**



# Πρόσθετο Υλικό

---

- Μελετήστε και τα παραδείγματα από το **Κεφάλαιο 13** του βιβλίου:  
«C++ How to Program, 9/e Paul & Harvey Deitel»  
[http://media.pearsoncmg.com/ph/esm/deitel/cpp\\_htp\\_9/code\\_examples/Code\\_Examples.zip](http://media.pearsoncmg.com/ph/esm/deitel/cpp_htp_9/code_examples/Code_Examples.zip)



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





# Σημείωμα Ιστορικού Εκδόσεων Έργου

---

Το παρόν έργο αποτελεί την έκδοση 1.0.1



# Σημείωμα Αναφοράς

---

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015.  
«Οντοκεντρικός Προγραμματισμός». Έκδοση: 1.0.1 Πάτρα 2015. Διαθέσιμο  
από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1105/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Διατήρηση Σημειωμάτων

---

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

# Σημείωμα Χρήσης Έργων Τρίτων

---

- Οι διαφάνειες βασίζονται στο βιβλίο «C++ How to Program, 8th Edition, Harvey M. Deitel, Paul J. Deitel, Prentice Hall.»

