



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Οντοκεντρικός Προγραμματισμός

Ενότητα 5: Η ΓΛΩΣΣΑ C++

**Εισαγωγή στην C++**

**ΔΙΔΑΣΚΟΝΤΕΣ:** Ιωάννης Χατζηλυγερούδης, Χρήστος Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

# Η Γλώσσα C++

# ΙΣΤΟΡΙΑ

---

1967: **Simula67** (Νορβηγία) -> πρώτη αντικειμενοστρεφής γλώσσα

'70: **Smalltalk** (Palo Alto, CA) -> κάθε στοιχείο ένα αντικείμενο  
Αρχές '80: ο αντικειμενοστρεφής τρόπος σκέψης εισάγεται σε ακαδημαϊκούς κύκλους

'80: **C++** (Stroustrup, AT&T): σοβαρή, αποδοτική γλώσσα, πρότυπο στη βιομηχανία

1995: **Java** (Sun Microsystems)



# ΔΙΑΦΟΡΕΣ ΜΕ C

---

## C

- Διαδικαστικός προγραμματισμός
- Από άνω προς τα κάτω, σχεδίαση προγράμματος με C

## C++

- Object based προγραμματισμός (κλάσεις, αντικείμενα, ενθυλάκωση)
- Object oriented προγραμματισμός (κληρονομικότητα, πολυμορφισμός)
- Generic πολυμορφισμός (class και function templates)



# COMPILERS

---

- CYGWIN (G++)
- Devc++
- Code::blocks



# ΕΙΣΟΔΟΣ ΕΞΟΔΟΣ

---

## ■ **cin**

- Standard input stream
- Συνήθως από το πληκτρολόγιο

## ■ **cout**

- Standard output stream
- Συνήθως έξοδος στην οθόνη

## ■ **cerr**

- Standard error stream
- Προβολή μηνυμάτων σφαλμάτων



# Απλό πρόγραμμα - Εκτύπωση

```
#include <iostream>
int main() {
    std::cout << "Hello World!\n";
    return 0;
}
```

Hello World!

`std::cout` είναι το κανάλι εξόδου στην οθόνη

Χρήση τελεστή `<<`

Χαρακτήρας escape: `/`

`/n` επόμενη γραμμή

`/t` κενό "tab"

`//` χαρακτήρας `/`

`/"` χαρακτήρας `"`

**Παρατήρηση:** Δεν είναι ανάγκη να βρίσκεται μέσα σε κάποια κλάση όπως στην Java



# Απλό πρόγραμμα – Είσοδος/Εξοδος

```
#include <iostream>
int main() {
    int num; std::cout << "Enter integer\n";
    std::cin >> num;
    std::cout << "Your number was: " << num << std::endl;
    return 0;
}
```

```
Enter integer
5
Your number was: 5
```

Διαδοχική χρήση του τελεστή <<

`std::in` είναι το κανάλι εισόδου από το πληκτρολόγιο

Χρήση τελεστή >>

Το `std::endl` μεταφέρει σε επόμενη γραμμή (αναγκάζει το κείμενο να εκτυπωθεί)





# Χρήση Using, Inline συναρτήσεις

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;

inline double diameter( const double r ) { return 2 * r; }

int main(){
    double r;
    for ( int k = 1; k < 3; k++ ) {
        cout << "Enter radius of circle: ";
        cin >> r;
        cout << "The diameter of the circle is "
             << diameter( r ) << endl;
    }
    return 0;
}
```

- Το `std::` στην αρχη μπορεί να παραληφθεί αν έχει γίνει πριν η δήλωση `using std::cout;`
- Συναρτήσεις με τον προσδιοριστή `inline`, “αναγκάζουν” τον compiler να ενσωματώσει τον κώδικα τους στο σημείο όπου καλούνται.  
(καλύτερη απόδοση σε ταχύτητα εκτέλεσης, αλλά αυξάνεται το μέγεθος του εκτελέσιμου)



# Πέρασμα με Αναφορά

```
#include <iostream>
using std::cout;
using std::endl;

int fooByValue ( int x) {
    x = x + 5;
    return x;
}

void fooByReference ( int &x ) {
    x = x + 5;
}

int main() {
    int number = 4;
    cout << "fooByValue returns: " << fooByValue(number) << endl;
    cout << "Number: " << number << endl; fooByReference(number);
    cout << "Number" << number << endl;
    return 0;
}
```

Περνάει στην συνάρτηση ένα αντίγραφο του ορίσματος (δεν αλλάζει η τιμή του number)

Περνάει το ίδιο το όρισμα (αύξηση του x σημαίνει αύξηση του number)

fooByValue returns: 9

Number: 4

Number9

Δεν άλλαξε η τιμή του **number** από την κληση της **fooByValue**

Άλλαξε η τιμή του **number** από την κληση της **fooByReference**



# Προκαθορισμένες Τιμές Ορισμάτων

Δήλωση προκαθορισμένων τιμών για τα ορίσματα συνάρτησης για την περίπτωση που παραληφθούν.

```
#include <iostream>
using std::cout;
using std::endl;

int foo( int a = 0, int b = 0, int c = 0 ){
    return a*100 + b* 10 + c; }

int main(){
    cout << "All default: " << foo() <<endl
         << "b,c default: " << foo( 1 ) <<endl
         << "c default: " << foo( 1,2 ) <<endl
         << "no default: " << foo( 1, 2, 3 ) <<endl;
    return 0;
}
```

```
All default: 0
b,c default: 100
c default: 120
no default: 123
```



# Πρόσθετο Υλικό

---

- Μελετήστε και τα παραδείγματα από το **Κεφάλαιο 2** του βιβλίου:  
«C++ How to Program, 9/e Paul & Harvey Deitel»  
[http://media.pearsoncmg.com/ph/esm/deitel/cpp\\_hpt\\_9/code\\_examples/Code\\_Examples.zip](http://media.pearsoncmg.com/ph/esm/deitel/cpp_hpt_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.»

