

# 343 – Εισαγωγή στον Προγραμματισμό

Τμήμα Μαθηματικών  
Πανεπιστήμιο Ιωαννίνων

Ακαδημαϊκό Έτος 2015-2016

Χάρης Παπαδόπουλος  
207δ, Β' όροφος  
e-mail: [charis@cs.uoi.gr](mailto:charis@cs.uoi.gr)

Ωρες Γραφείου:  
Πέμπτη 11-13

# Στόχος Μαθήματος

- Ανάλυση και επίλυση προβλημάτων με Η/Υ
- Γλώσσα προγραμματισμού: C/C++
  - Δυαδική αναπαράσταση
  - Είσοδος/Εξοδος δεδομένων, τύποι δεδομένων & μεταβλητών
  - Αριθμητικοί και λογικοί τελεστές
  - Ροή ελέγχου προγράμματος: if/else, switch, for, while, do-while
  - Συναρτήσεις, εμβέλεια μεταβλητών και αναδρομή
  - Πίνακες (μονοδιάστατοι και πολυδιάστατοι)
  - Ψευδοτυχαίοι αριθμοί και Συμβολοσειρές
  - Χρήση αρχείων
  - Εγγραφές και δομές
  - Εφαρμογές σε ταξινομήσεις και αναζήτηση στοιχείων
- Διαλέξεις (3 ώρες) και Εργαστήρια (2 ώρες)
- Η συμμετοχή στις εργαστηριακές ασκήσεις είναι υποχρεωτική

# Διδασκαλία

- Κάθε εβδομάδα
  - 3ωρη διάλεξη: ανάλυση χαρακτηριστικών της C/C++
  - 2ωρο εργαστήριο: πρακτική εξάσκηση (υλοποίηση προγραμμάτων) μέσα από συγκεκριμένα προβλήματα
- Εργαστήρια:
  - 7 εβδομαδιαία εργαστήρια (5+2Quiz)
  - Σκοπός των εργαστηρίων είναι η εμπέδωση της ύλης, η εξάσκηση και η λύση αποριών
  - Η παρακολούθηση είναι υποχρεωτική και είναι το *διαβατήριο* για να λάβετε μέρος στις τελικές εξετάσεις
  - Μπορείτε να λείψετε σε **1 το πολύ** εργαστήριο
  - Προσοχή: Εάν λείψετε σε παραπάνω εργαστήρια δεν μπορείτε να λάβετε μέρος στις εξετάσεις !!!

# Εργαστήρια & Τεστ Quiz

- Στα Εργαστήρια θα πρέπει:
  - Να κατανοείτε το πρόβλημα και να προσπαθείτε να σχεδιάσετε την λύση **πριν έλθετε στο εργαστήριο**. → PreLab.pdf
  - Να συμμετέχετε ενεργά και να είστε **προετοιμασμένοι** να απαντάτε σε ερωτήσεις σχετικές με την εκφώνηση → Lab.pdf
- Θα βαθμολογηθείτε με **ερωτήσεις κουίζ** σε 2 εργαστήρια με βαθμολογία 20% του τελικού βαθμού.
  - Θέματα πολλαπλών επιλογών διαλεγμένα από θεωρία και εργαστήρια.
- **Ερωτήσεις Κουίζ:**
  - Αποτελούν ενδιάμεσα τεστ που βαθμολογούν την επίδοσή σας
  - Πολύ πιο δύσκολα από ό,τι φαντάζεστε... (...αν δεν μελετάτε συστηματικά)
  - Μετά από ~3 εργαστήρια θα εφαρμόζονται (...αν απουσιάζετε, δεν βαθμολογείστε).

# Τρόπος Εξέτασης Quiz

- Ερωτήσεις Κουίζ:

<http://ecourse.uoi.gr>

Μαθηματικών -> Προπτυχιακά Μαθήματα

- Θα πρέπει να δημιουργήσετε ΝΕΟ ΛΟΓΑΡΙΑΣΜΟ (εκτός αν ήδη έχετε) και θα σας σταλεί μέσω email κωδικός για το μάθημα
- Παλαιότεροι φοιτητές: όσοι έχουν ήδη παρακολουθήσει επιτυχώς τα εργαστήρια μπορούν να προσέλθουν στις τελικές εξετάσεις

# Τελική Βαθμολογία

- **Επιτυχής παρακολούθηση** των υποχρεωτικών εργαστηρίων
  - μία (1) το πολύ απουσία σε **πέντε (5)** εργαστηριακά μαθήματα
- **Δυο (2) τεστ quiz (Q1, Q2)** με βάρος 10% του συνολικού βαθμού το καθένα.
  - Όσοι απουσιάσουν σε ένα τεστ quiz: μηδενίζεται το 10%
- **Παλαιότεροι φοιτητές:** μπορούν να λάβουν μέρος στα τεστ quiz (με την προϋπόθεση ότι έχουν ήδη παρακολουθήσει επιτυχώς τα εργαστήρια)
- **Τελική βαθμολογία** (με επιτυχή παρακολούθηση εργαστηρίων):

$$TB = \max \left\{ \begin{array}{l} (10\% Q1) + (10\% Q2) + (80\% \Gamma\rho E\xi) + A. E. \\ \Gamma\rho E\xi \end{array} \right\}$$

# Χρήσιμο Υλικό

- Σελίδα Μαθήματος:

<http://www.cs.uoi.gr/~charis/c343>

- Διαφάνειες Διαλέξεων
- Εργαστηριακές Ασκήσεις
- Υλοποιημένα προγράμματα και παραδείγματα
- Ανακοινώσεις

- Λογισμικό υλοποίησης προγραμμάτων :

Bloodshed DevC++ :



<http://www.bloodshed.net/>



Τμήμα Μηθματικών  
Γραφείο: 2076  
Ώρες Γραφείου: Δευτέρα 11 - 13 και Παρασκευή 11 - 13  
Ώρες Μαθημάτων: Διαλέξεις: Πέμπη 09-12 Εργαστήρια: Τρίτη 14:00-20:00

343 Εισαγωγή στον Προγραμματισμό  
Ακαδημαϊκό Έτος 2013-2014

Τηλ. Γρ.: 26510 - 08224  
E-mail: [charis@cs.uoi.gr](mailto:charis@cs.uoi.gr)

http: [www.cs.uoi.gr/~charis/c343](http://www.cs.uoi.gr/~charis/c343)  
[course.uoi.gr/course/view.php?id=xxxxx](http://course.uoi.gr/course/view.php?id=xxxxx)

012 και  
Εργαστ. Η'Υ, 1<sup>ο</sup> ορόφου, Αναγνωστήριο  
Εργαστ. Η'Υ, 1<sup>ο</sup> ορόφου, Μικρό Ανασαστή

• Αρχή • Χρήσιμο Υλικό • Διαφάνειες • Εργαστήρια • Ημερολόγιο • Ανακοινώσεις

## Περιληψη

Βασικά χαρακτηριστικά της γλώσσας προγραμματισμού C++. Σχεδίαση και ανάλυση υπολογιστικών προγραμμάτων, διάθωση σχολίων, έλεγχος τεκμηρίωση, εγχειρίδιο χρήσης και συντηκός προγραμματισμός, βασικοί τύποι δεδομένων, εντολές έλεγχου ροής προγράμματος, είσοδος δεδομένων και έξοδος αποτελεσμάτων. Τύποι δεδομένων, συμβολοσειρές, και πίνακες.

Υποπρογράμματα, βασικές και αναδρομικές συναρτήσεις, διαβίβαση τιμών των παραμέτρων δια μέσο τιμής και δια μέσο διεύθυνσης. Διάφορα (ζωή προσδιοριστών και κανόνες εμφάνισης και ορατότητας, χρήση αρχείων, Δομές, εγγραφές, λίστες και σύνθετοι τύποι δεδομένων. Προεπεξεργαστής κτ βιβλιοθήκες. Τεχνικές αποτελεσματικού προγραμματισμού και εξοικείωση με αλγόριθμους και διαγράμματα ροής προγραμμάτων. Εφαρμογές σε προβλήματα αναζήτησης, ταξινόμησης και μαθηματικών προβλημάτων.

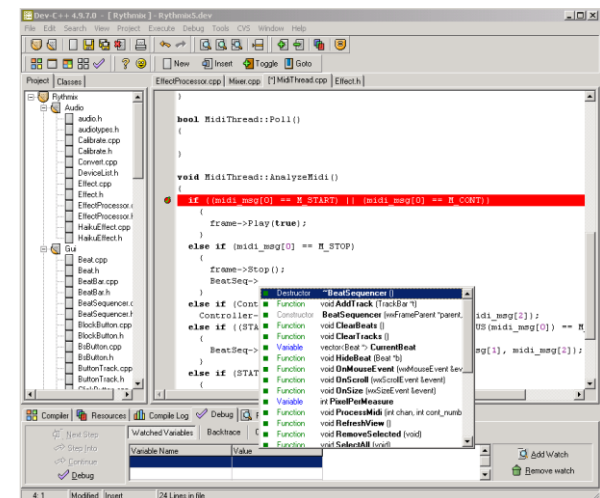
Στο μάθημα περιλαμβάνονται εργαστηριακές ασκήσεις στις οποίες η συμμετοχή είναι υποχρεωτική.

## Βιβλιογραφία

- [1] W. Savitch, Πλήρης C++, Εκδόσεις Τρίδα, 2011. Κωδικός Ευδ: 18548892
- [2] H. Deitel and P. Deitel, C++ Προγραμματισμός 4η Έκδοση, Εκδόσεις Μ. Γκουρβας, 2013. Κωδικός Ευδ: 12336819
- [3] L. Jesse, Πλήρης εγχειρίδιο της C++, Εκδόσεις Α. Γκουρβας, 2006. Κωδικός Ευδ: 12374
- [4] Ν. Χατζηγιαννακός, Η γλώσσα C++ σε βάθος, Εκδόσεις Κλειδαρίτσας, 2008. Κωδικός Ευδ: 13761

## Τρόπος Βαθμολόγησης

- Προϋπόθεση για την δυνατότητα βαθμολόγησης είναι η επιτυχής παρακολούθηση των υποχρεωτικών εργασιών.
  - ο Επιτυχής παρακολούθηση = μιο (1) το πολύ απουσία στο 66 (6) εργαστηριακά μαθήματα



# Τμήματα Εργαστηρίων

- **Εργαστήριο (μεγάλο):** Εργαστήριο Η/Υ 1<sup>ο</sup> όροφο δίπλα από το Αναγνωστήριο
- Τα εργαστήρια θα ξεκινήσουν **Δευτέρα 19 Οκτωβρίου**

## Εργαστήριο Δευτέρα (14:00-20:15)

A1 Δευτέρα <b>14:00-15:15</b>	_____ - 10636
A2 Δευτέρα <b>15:15-16:30</b>	10637 - 10702
A3 Δευτέρα <b>16:30-17:45</b>	10703 - 10786
A4 Δευτέρα <b>17:45-19:00</b>	10787 - 10881
A5 Δευτέρα <b>19:00-20:15</b>	10882 - _____

## Αλλαγές ΔΕΝ επιτρέπονται!

- Διάρκεια Εργαστηρίου: 1h:15m



Θ: διάλεξη (θεωρία)

Ε: Εργαστήριο

Q: Τεστ quiz

# Ημερολόγιο Μαθήματος

## Οκτώβριος 2015

Δ	Τ	Τ	Π	Π
			1	2
5	6	7	8	9 Θ
12	13	14	15	16 Θ
19 Ε	20	21	22	23 Θ
26	27	28	29	30 Θ

## Νοέμβριος 2015

Δ	Τ	Τ	Π	Π
2 Ε	3	4	5	6 Θ
9 Ε	10	11	12	13 Θ
16 Q	17	18	19	20 Θ
23 Ε	24	25	26	27 Θ
30 Ε				

## Δεκέμβριος 2015

Δ	Τ	Τ	Π	Π
	1	2	3	4 Θ
7 Q	8	9	10	11 Θ
14	15	16	17	18 Θ

## Ιανουάριος 2016

Δ	Τ	Τ	Π	Π
4	5	6	7	8
11	12	13	14	15 Θ

Εβδομάδα	Θέματα	Υψη βιβλιογραφίας
Πα, 9 Οκτωβρίου	Εισαγωγικά μαθήματος & Δυαδική αναπαράσταση	[1]: 1.1, Παράρτημα 3 [2]: Κεφ. 1, Β, Δ
Πα, 16 Οκτωβρίου	Είσοδος/Εξοδος δεδομένων, τύποι δεδομένων & μεταβλητών	[1]: 1.2, 1.3, 1.4, 1.5, Παράρτημα 1 [2]: Κεφ. 2, Γ
Δε, 19 Οκτ	1 <sup>ο</sup> Εργαστήριο	
Πα, 23 Οκτωβρίου	Προεπεξεργαστής, αριθμητικοί και λογικοί τελεστές	[1]: 2.1, Παράρτημα 2 [2]: 4.11, 4.12, Α, ΣΤ
Πα, 30 Οκτωβρίου	Ροή ελέγχου: if/else, switch, for, while, do-while και ροή ελέγχου if/else	[1]: 2.2, 2.3 [2]: Κεφ. 4, Κεφ. 5
Δε, 2 Νοε	2 <sup>ο</sup> Εργαστήριο	
Πα, 6 Νοεμβρίου	Συναρτήσεις, εμβέλεια μεταβλητών και αναδρομή	[1]: 3.1, 3.2, 3.3, 4.1, 4.2, 13.1, 13.2 [2]: Κεφ. 6
Δε, 9 Νοε	3 <sup>ο</sup> Εργαστήριο	
Πα, 13 Νοεμβρίου	Επανάληψη με Παραδείγματα	[1]: 5.1, 5.2, 5.4 [2]: Κεφ. 7
Δε, 16 Νοε	1 <sup>ο</sup> Quiz	
Πα, 20 Νοεμβρίου	Πίνακες (μονοδιάστατοι και πολυδιάστατοι)	[1]: 5.1, 5.2, 5.4 [2]: Κεφ. 7
Δε, 23 Νοε	4 <sup>ο</sup> Εργαστήριο	
Πα, 27 Νοεμβρίου	Εφαρμογές σε ταξινομήσεις και αναζήτηση στοιχείων	[1]: Παράρτημα 4, 9.1, 9.2, 9.3 [2]: 6.7, 6.8, Κεφ. 18
Δε, 30 Νοε	5 <sup>ο</sup> Εργαστήριο	
Πα, 4 Δεκεμβρίου	Αλφαριθμητικά και Συμβολοσειρές	[1]: 6.1, 12.1, 12.2, 12.4 [2]: Κεφ. 21, 17.1-17.10
Δε, 7 Δεκ	2 <sup>ο</sup> Quiz	
Πα, 11 Δεκεμβρίου	Εγγραφές, δομές και χρήση αρχείων	[1]: 5.3, 13.3 [2]: 7.7, 7.8, 8.6, Κεφ. 19
Πα, 18 Δεκεμβρίου	Επανάληψη	
Πα, 15 Ιανουαρίου	Επανάληψη	

# Βιβλιογραφία



[1] W. Savitch, Πλήρης C++,  
Εκδόσεις Τζιόλα, 2011.



[2] H. Deitel and P. Deitel, C++ Προγραμματισμός 6η Εκδοση,  
Εκδόσεις Μ. Γκιούρδας, 2013.

[3] L. Jesse, Πλήρες εγχειρίδιο της C++,  
Εκδόσεις Α. Γκιούρδα, 2006.

[4] Ν. Χατζηγιαννάκης, Η γλώσσα C++ σε βάθος,  
Εκδόσεις Κλειδάριθμος, 2008.

- Θα συμβούλευα να ακολουθήσετε τα [1] και [2]  
(και ειδικότερα το [1])
  - Σε αυτά αναφέρονται αναλυτικά οι αντίστοιχες ενότητες που καλύπτουμε

Ενότητα 1

# **ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ & ΤΟ ΠΡΩΤΟ ΠΑΡΑΔΕΙΓΜΑ**

# Εισαγωγή

- Υπολογιστής: συσκευή που μπορεί
  - να εκτελέσει υπολογισμούς και
  - να πάρει λογικές αποφάσεις, με
- ταχύτητες δισεκατομμύρια φορές μεγαλύτερες από εκείνες των ανθρώπων
- Αλλά για να ξέρει τι να κάνει ένας υπολογιστής, πρέπει κάποιος (εμείς) να του δώσει οδηγίες

Στο μάθημα αυτό θα μάθουμε να δίνουμε οδηγίες σε έναν υπολογιστή!

- Στόχος μας:
  - Διαφάνεια του προγράμματος, χρησιμοποιώντας δομημένο προγραμματισμό

# Προβλήματα

- **Πρόβλημα:** Μία κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της οποίας δεν είναι γνωστή, ούτε προφανής

# Κατηγορίες Προβλημάτων

- Με βάση την δυνατότητα επίλυσης:
  - **Επιλύσιμα:** Η λύση τους είναι γνωστή ή μπορεί να αποδειχθεί ότι έχουν λύση
  - **Ανοικτά:** Δεν υπάρχει διατυπωμένη λύση αλλά δεν έχει αποδειχθεί ότι δεν λύνονται
  - **Άλυτα:** Έχει αποδειχθεί ότι δεν έχουν λύση
- Με βάση το ζητούμενο:
  - **Απόφασης:** Το ζητούμενο του προβλήματος είναι η λήψη μιας απόφασης (π.χ. Αν ο αριθμός 11 είναι πρώτος ή όχι)
  - **Υπολογιστικά:** Το ζητούμενο είναι το αποτέλεσμα μιας σειράς υπολογισμών (π.χ. Να βρεθεί η δόση για πενταετές δάνειο 40000€ με επιτόκιο 4,82%)
  - **Βελτιστοποίησης:** Το ζητούμενο είναι η εύρεση ενός αποτελέσματος που ελαχιστοποιεί ή μεγιστοποιεί κάποια κριτήρια (π.χ. Βρες το πλησιέστερο βενζινάδικο)

# Προβλήματα και Η/Υ

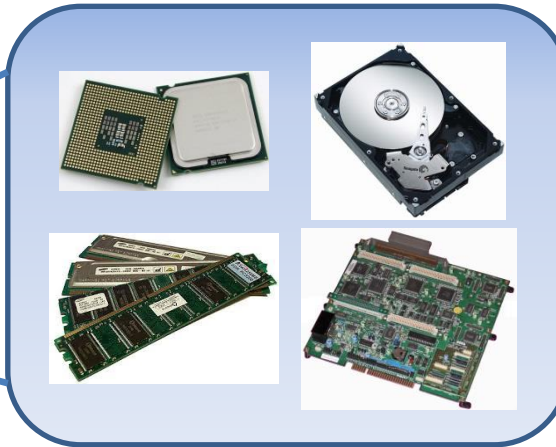
- Λόγοι επίλυσης προβλημάτων μέσω Η/Υ:
  - Ταχύτητα εκτέλεσης πράξεων  
(π.χ. Ένας Η/Υ με επεξεργαστή στα 3 GHz μπορεί να εκτελεί 3,000,000,000 απλές πράξεις ανά δευτερόλεπτο)
  - Μεγάλο πλήθος δεδομένων
  - Πολυπλοκότητα υπολογισμών
  - Επαναληπτικότητα διαδικασιών
- Πρωταρχικές λειτουργίες Η/Υ
  - Πρόσθεση: βασική αριθμητική πράξη
  - Σύγκριση: βασική λογική πράξη
  - Μεταφορά δεδομένων

# Η Ανατομία ενός Υπολογιστή

**Συσκευές Εξόδου:** οθόνη,  
εκτυπωτής, κ.τ.λ



**Συσκευές Εισόδου:** πληκτρολόγιο,  
ποντίκι, κ.τ.λ



**μονάδα υπολογιστή:**

- κεντρική μονάδα επεξεργασίας (CPU),
- μονάδα μνήμης,
- μονάδα δευτερεύουσας αποθήκευσης (π.χ σκληρός δίσκος),
- αριθμητική και λογική μονάδα (συνήθως μέσα στη CPU),
- δίαυλος επικοινωνίας (bus)

...



# Λογισμικό

Ο υπολογιστής χρειάζεται και οδηγίες για το τι να κάνει. Αυτό το ρόλο παίζουν τα διάφορα προγράμματα:

- **BIOS:** πρόγραμμα που βοηθάει τον υπολογιστή να ξεκινήσει
- **λειτουργικό σύστημα:** πρόγραμμα που παίρνει τον έλεγχο μετά την εκκίνηση και διαχειρίζεται όλους τους πόρους του υπολογιστή, τις συσκευές εισόδου και εξόδου και γενικά φροντίζει για την εκτέλεση άλλων προγραμμάτων  
π.χ. Windows, Linux, SunOS, MacOSX, κ.τ.λ
- **εφαρμογές γενικής χρήσης:** προγράμματα φτιαγμένα με συγκεκριμένο σκοπό,  
π.χ. αναπαραγωγή ταινιών, επεξεργασία εικόνας, κειμενογράφοι, παιχνίδια, κ.τ.λ

# Αλγόριθμοι

- **Αλγόριθμος:** Πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που επιλύουν ένα πρόβλημα
- Χαρακτηριστικά Αλγορίθμων:
  - **Είσοδος (Input):** Δεδομένα που δίνονται στον αλγόριθμο
  - **Έξοδος (Output):** Δεδομένα που ο αλγόριθμος παράγει ως αποτέλεσμα
  - **Καθοριστικότητα:** Κάθε εντολή πρέπει να καθορίζεται με πλήρη σαφήνεια
  - **Αποτελεσματικότητα:** Κάθε εντολή πρέπει να είναι απλή και να εκτελείται σε εύλογο χρονικό διάστημα
  - **Περατότητα:** Ο αλγόριθμος πρέπει να τερματίζει μετά από πεπερασμένο αριθμό βημάτων

# Αναπαράσταση Αλγορίθμων

- Ένας αλγόριθμος μπορεί να αναπαρασταθεί με διάφορους τρόπους
  - Ελεύθερο κείμενο
    - Ανεπεξέργαστος και αδόμητος τρόπος παρουσίασης
    - Δεν εξασφαλίζει ούτε την καθοριστικότητα ούτε την αποτελεσματικότητα
  - Διαγραμματικές Τεχνικές
    - Γραφικός τρόπος παρουσίασης
    - Ο πιο συνηθισμένος είναι με τα διαγράμματα ροής
  - Φυσική γλώσσα κατά βήματα
    - Π.χ. Ψευδοκώδικας
    - Χρειάζεται προσοχή για να μην παραβιαστεί το κριτήριο της καθοριστικότητας
  - **Κωδικοποίηση**
    - **Συγγραφή του αλγορίθμου σε μια γλώσσα προγραμματισμού**

# Πληροφορική

- **Πληροφορική:** Η επιστήμη που μελετά τους αλγορίθμους από τις ακόλουθες σκοπιές:
  - **Υλικού (Hardware):** Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες των τμημάτων του υλικού
  - **Γλωσσών Προγραμματισμού:** Το είδος της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί επηρεάζει την αποδοτικότητα του αλγορίθμου
  - **Θεωρητική:** Έρευνα θεωρητικών θεμάτων όπως ή ύπαρξη ή όχι λύσης για ένα πρόβλημα
  - **Αναλυτική:** Μελέτη του πλήθους των υπολογιστικών πόρων που απαιτούνται από έναν αλγόριθμο

# Προγράμματα & Προγραμματισμός Η/Υ

- **Πρόγραμμα Η/Υ (Computer Program):** Μια οργανωμένη λίστα οδηγιών που όταν εκτελείται από έναν υπολογιστή, αυτός συμπεριφέρεται με έναν συγκεκριμένο προσχεδιασμένο τρόπο
- **Προγραμματισμός Η/Υ (Computer Programming):** Η διαδικασία της δημιουργίας ενός προγράμματος που επιλύει ένα συγκεκριμένο πρόβλημα.  
Ο Προγραμματισμός περιλαμβάνει δύο βήματα:
  - Σχεδιασμός της λογικής που επιλύει το πρόβλημα
  - Κωδικοποίησης της λογικής σε μια γλώσσα προγραμματισμού
- **Γλώσσες Προγραμματισμού (Programming Languages):** Τεχνητές γλώσσες κωδικοποίησης προγραμμάτων Η/Υ
  - Μεγάλη σαφήνεια και ακρίβεια
  - Γεφυρώνουν το χάσμα ανάμεσα στις γλώσσες Η/Υ και ανθρώπων

# Υλοποίηση Προγραμμάτων

- Βήματα υλοποίησης ενός προγράμματος:



- Συγγραφή πηγαίου κώδικα (source code)

- Δημιουργία αρχείου κειμένου (.cpp) που περιέχει τις εντολές του προγράμματος
- Δημιουργείται είτε σε οποιοδήποτε κειμενογράφο (π.χ. Wordpad, Textpad) ή στον συντάκτη (editor) ολοκληρωμένων περιβαλλόντων ανάπτυξης (IDE), όπως είναι το DevC++

- Μεταγλώττιση (compilation) πηγαίου κώδικα

- Διαδικασία μετάφρασης του πηγαίου κώδικα σε κώδικα μηχανής
- Υλοποιείται από εξειδικευμένα προγράμματα (μεταγλωττιστές-compilers)
- Γίνεται έλεγχος για συντακτικά και γραμματικά λάθη στον πηγαίο κώδικα

- Εκτέλεση (execution) για έλεγχο καλής λειτουργίας

# Τι καταλαβαίνει ένας υπολογιστής;

Ένας υπολογιστής καταλαβαίνει από 0 και από 1

- Αλλά ένας άνθρωπος δε μπορεί εύκολα να μιλάει στον υπολογιστή με 0 και 1

π.χ

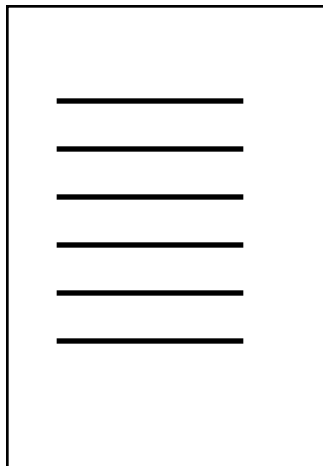
000000 00001 00010 00110 00000 100000

- σε κάποια αρχιτεκτονική λέει στον υπολογιστή να προσθέσει τους καταχωρητές 1 και 2 και να βάλει το αποτέλεσμα τον καταχωρητή 6.

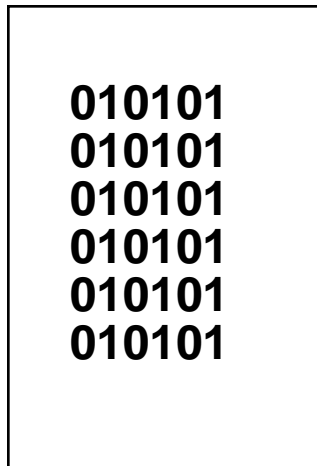
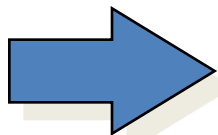
# Μεταγλώττιση & Εκτέλεση

μετάφραση

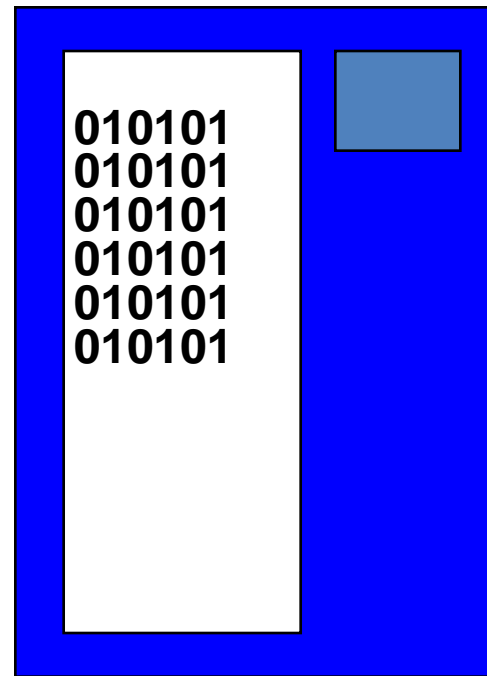
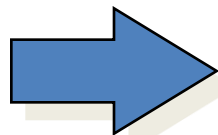
εκτέλεση



Πηγαίος κώδικας



Κώδικας μηχανής



υπολογιστής



# Από τις Γλώσσες Μηχανής μέχρι σήμερα

- γλώσσα μηχανής: 1000 10000 10001
- συμβολομεταφραστικές γλώσσες (assembly language):  
συντομεύσεις της αγγλικής γλώσσας,

π.χ. ADD 32 6, 1, 2

Ένα πρόγραμμα, ο assembler αναλαμβάνει να μετατρέψει την παραπάνω εντολή σε γλώσσα μηχανής

- γλώσσες υψηλού επιπέδου: οδηγίες που μοιάζουν σχεδόν όπως τα καθομιλουμένα αγγλικά και περιέχουν συνήθη μαθηματικά σύμβολα,

π.χ.  $x6 = x1 + x2;$

Ένα ακόμη πρόγραμμα ο μεταγλωττιστής (compiler) αναλαμβάνει να μετατρέψει το πρόγραμμα υψηλού επιπέδου σε συμβολομεταφραστική γλώσσα.

# Προέλευση της C/C++ (Ιστορικά)

- C++: γλώσσα προγραμματισμού C με κλάσεις (αντικείμενα)
- Ιστορικά της C:
  - 1969-1973 AT & T Bell Labs, Dennis Ritchie
  - 1978 “The C Programming Language”, Kernighan & Ritchie
  - 1989-1990 Αποδοχή ANSI/ISO Standard (ANSI C)
- Δημοφιλή προγράμματα:
  - C μαζί με το λειτουργικό σύστημα UNIX
- Εξέλιξη σε C++:
  - 1980: AT & T Bell Labs, Bjarne Stroustrup
  - C++ : (σχεδόν) υπερσύνολο της C
  - Αντικειμενοστραφή γλώσσα προγραμματισμού
- Στο μάθημα: θα δούμε την C++ ως μια προχωρημένη γλώσσα προγραμματισμού C αλλά χωρίς αντικείμενα (C/C++)

# Χαρακτηριστικά

- Κύρια συστατικά:
  - συναρτήσεις
  - κλάσεις (αντικείμενα)
- Δομημένη γλώσσα
- Χαμηλού και υψηλού επιπέδου προγραμματισμός
  - Ο προγραμματιστής έχει πλήρη έλεγχο και ευθύνεται για τυχόν σφάλματα του
- Στάνταρ Βιβλιοθήκη της C++ (Standard Library)
  - Εκμάθηση της γλώσσας C++ (δημιουργία νέων συστατικών που τοποθετούνται στην βιβλιοθήκη για περαιτέρω χρήση)
  - Χρήση των ήδη υλοποιημένων κλάσεων και συναρτήσεων που υπάρχουν στη βιβλιοθήκη

# Δομημένος Προγραμματισμός

- Οργάνωση του προγράμματος σε μικρές, αυτόνομα κατανοητές, οντότητες
  - Κάθε τέτοια οντότητα έχει ένα σημείο εισόδου και ένα σημείο εξόδου
- Η δομή ενός προγράμματος βασίζεται στις τρεις παρακάτω λογικές δομές:
  1. ακολουθία (sequence)
  2. επιλογή (condition)
  3. επανάληψη (repetition)
- Με τις παραπάνω δομές μπορεί να εκφραστεί *οποιοσδήποτε αλγόριθμος*.

# Το πρώτο παράδειγμα

```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

- Εκτυπώνει στην οθόνη (τερματικό παράθυρο):

Hello !!

Ας μάθουμε C και C++

# Επεξήγηση του παραδείγματος

```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

- Ένα πρόγραμμα αποτελείται από την κύρια συνάρτηση (`main`)
- Εκτέλεση προγράμματος  $\Rightarrow$  Εκτέλεση της `main`
- Κυρίως σώμα (εντολές) της `main`: περικλείεται από `{ }`

# Επεξήγηση του παραδείγματος



```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

- Η εντολή `#include` ανοίγει κάποια βιβλιοθήκη έτοιμων συν/σεων
  - Η βιβλιοθήκη `iostream` περιέχει συναρτήσεις εισόδου και εξόδου
  - Θα χρειαστούμε την συνάρτηση `cout`

# Επεξήγηση του παραδείγματος



```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

- Κενές γραμμές: Δεν επηρεάζουν το πρόγραμμα (αγνοούνται από τον μεταφραστή)
- Κάνουν το πρόγραμμα πιο

ευανάγνωστο!



# Επεξήγηση του παραδείγματος



```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

- Χώροι ονομάτων: κάποια κομμάτια κώδικα ανήκουν σε περιοχές ονομάτων (namespace)
  - βοηθούν στην διαχείριση μεγάλων προγραμμάτων και βιβλιοθηκών ταυτόχρονα
  - μεταβλητές ή και συναρτήσεις με το ίδιο όνομα αλλά σε διαφορετικό namespace
- Χρήση της συνάρτησης `cout` που βρίσκεται στο `std namespace`

# Επεξήγηση του παραδείγματος

```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

- Η `main` είναι μια *συνάρτηση*
  - χωρίς παραμέτρους ()
  - επιστρέφει μια τιμή `int` (ακέραια)
- Το κυρίως σώμα της `main` περικλείεται από { }

# Επεξήγηση του παραδείγματος

```
#include <iostream>

using namespace std;

int main()
{
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;
}
```



- Χρήση της συνάρτησης cout για χειρισμό εξόδου:
  - Εμφανίζει το κείμενο των " " στην οθόνη: \_Hello !!
  - \n : δεν εκτυπώνεται ως "\n", είναι χαρακτήρας αλλαγής γραμμής και δίνει εντολή να ξεκινήσει μια νέα γραμμή στην έξοδο

# Επεξήγηση του παραδείγματος

```
#include <iostream>


using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```



- Χρήση της συνάρτησης cout για χειρισμό εξόδου:
  - Εμφανίζει το κείμενο των " " στην οθόνη: Ας μάθουμε C και C++
  - \n : δεν εκτυπώνεται ως "\n", είναι χαρακτήρας αλλαγής γραμμής και δίνει εντολή να ξεκινήσει μια νέα γραμμή στην έξοδο

# Επεξήγηση του παραδείγματος


```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;
}
```



- Το πρόγραμμα σταματάει την εκτέλεση.
- Τερματίζει την συνάρτηση `main` και ως τιμή της συνάρτησης `main` επιστρέφει τον ακέραιο αριθμό 0

# Στην πράξη;

```
#include <iostream>

using namespace std;

int main()
{

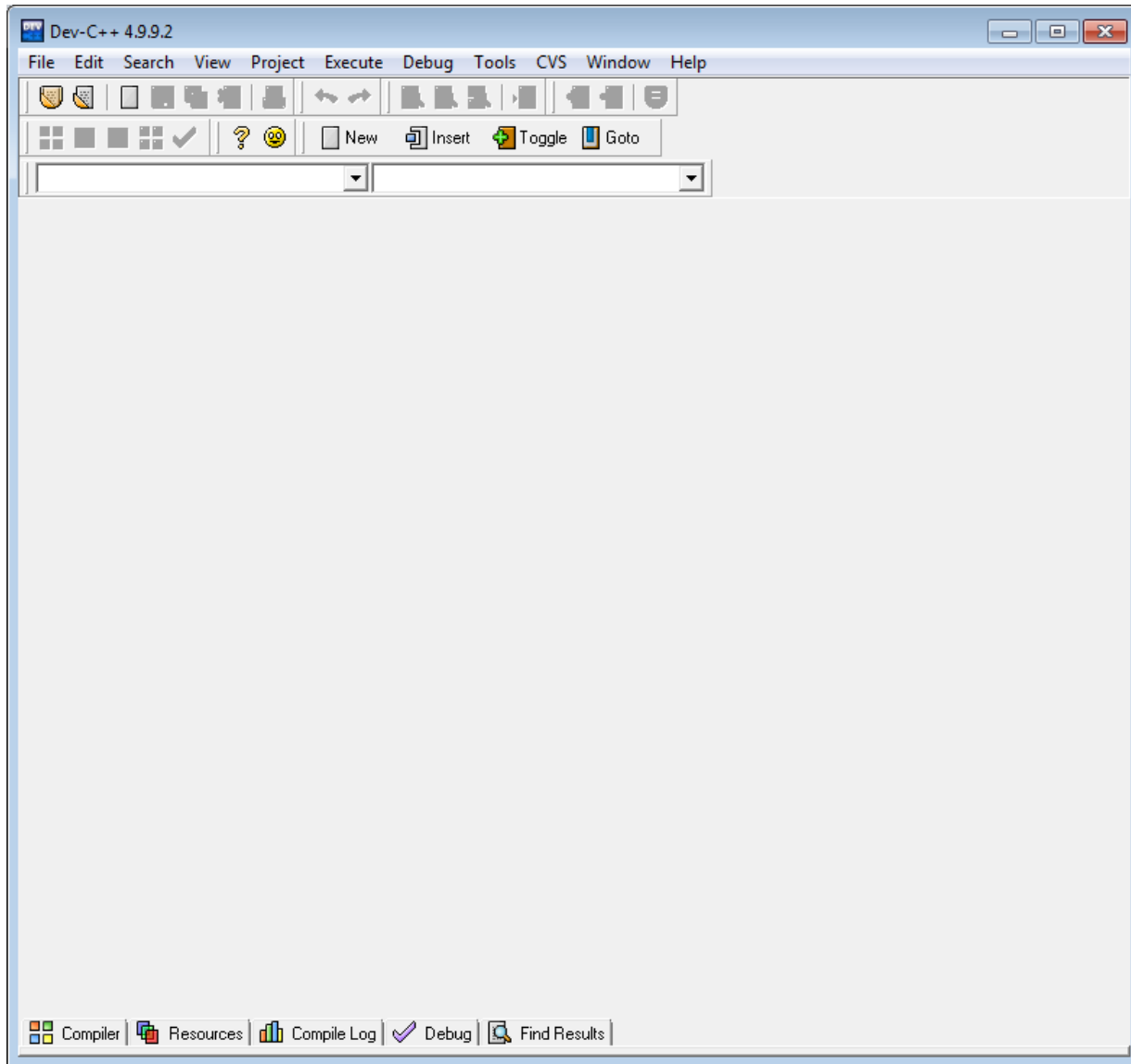
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

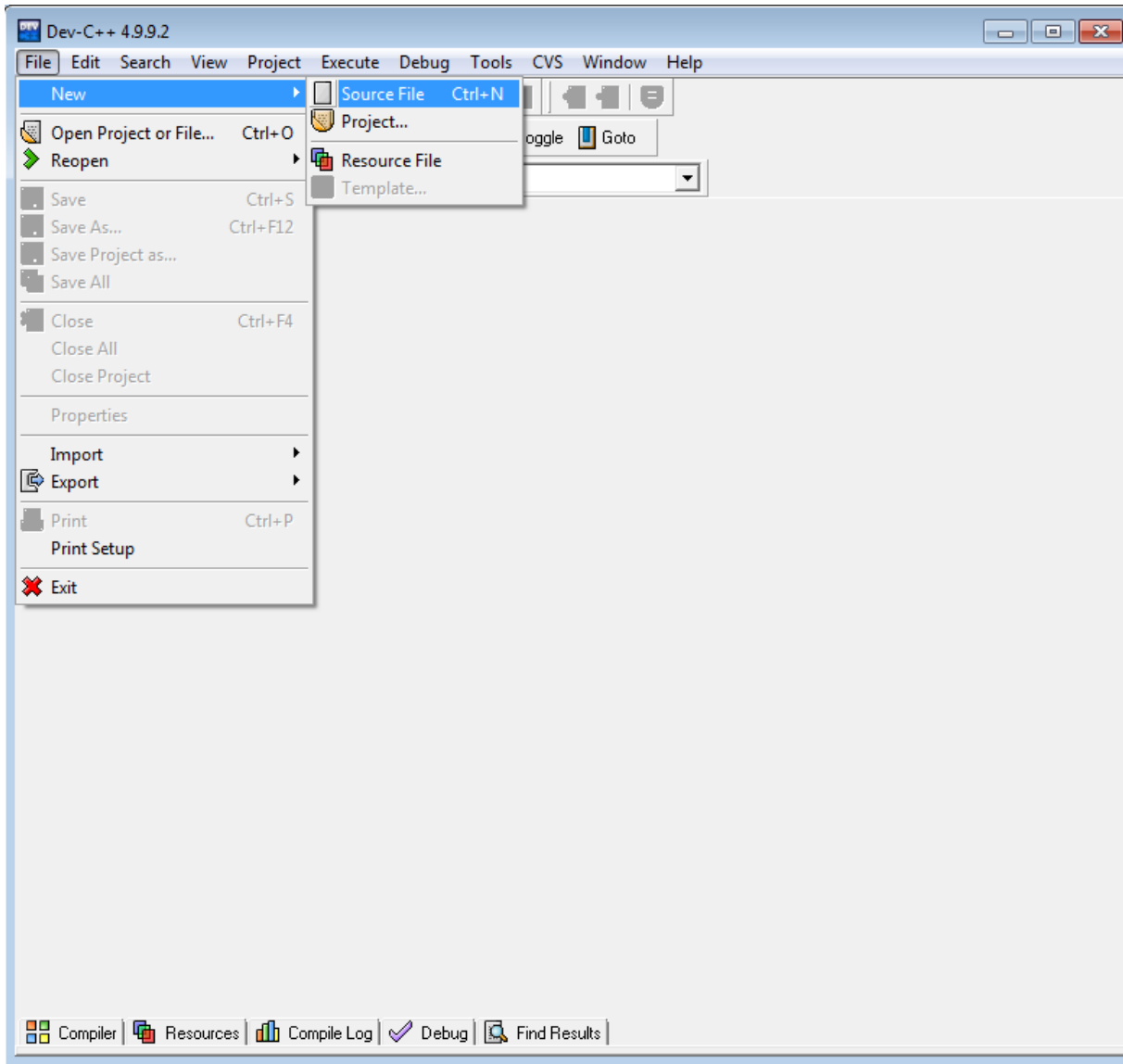
- Ας δούμε πώς υλοποιούμε το άνω πρόγραμμα στην πράξη
  - Θα χρειαστείτε την διαδικασία αυτή πολλές φορές στο εργαστήριο
- Δεν είναι τόσο απλά...

# Στην πράξη;



- Ανοίγουμε το devc++

# Στην πράξη;



- Δημιουργούμε ένα νέο αρχείο (source file)



# Στην πράξη;

A screenshot of the Dev-C++ 4.9.9.2 IDE. The window title is "Dev-C++ 4.9.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, editing, and debugging. The main editor area shows a C++ program in a file named "Untitled1". The code is as follows:

```
#include <iostream>

using namespace std;

int main()
{
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;
}
```

The second line of the cout statement is highlighted in light blue. The status bar at the bottom shows icons for Compiler, Resources, Compile Log, Debug, and Find Results.

- Γράφουμε τον κώδικα

# Στην πράξη;



- Αποθηκεύουμε το αρχείο (με κατάληξη .cpp)

A screenshot of the Dev-C++ 4.9.2 IDE. The window title is "Dev-C++ 4.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. Below the toolbar is a dropdown menu. The main editor area shows a C++ program named "example.cpp" with the following code:

```
#include <iostream>

using namespace std;

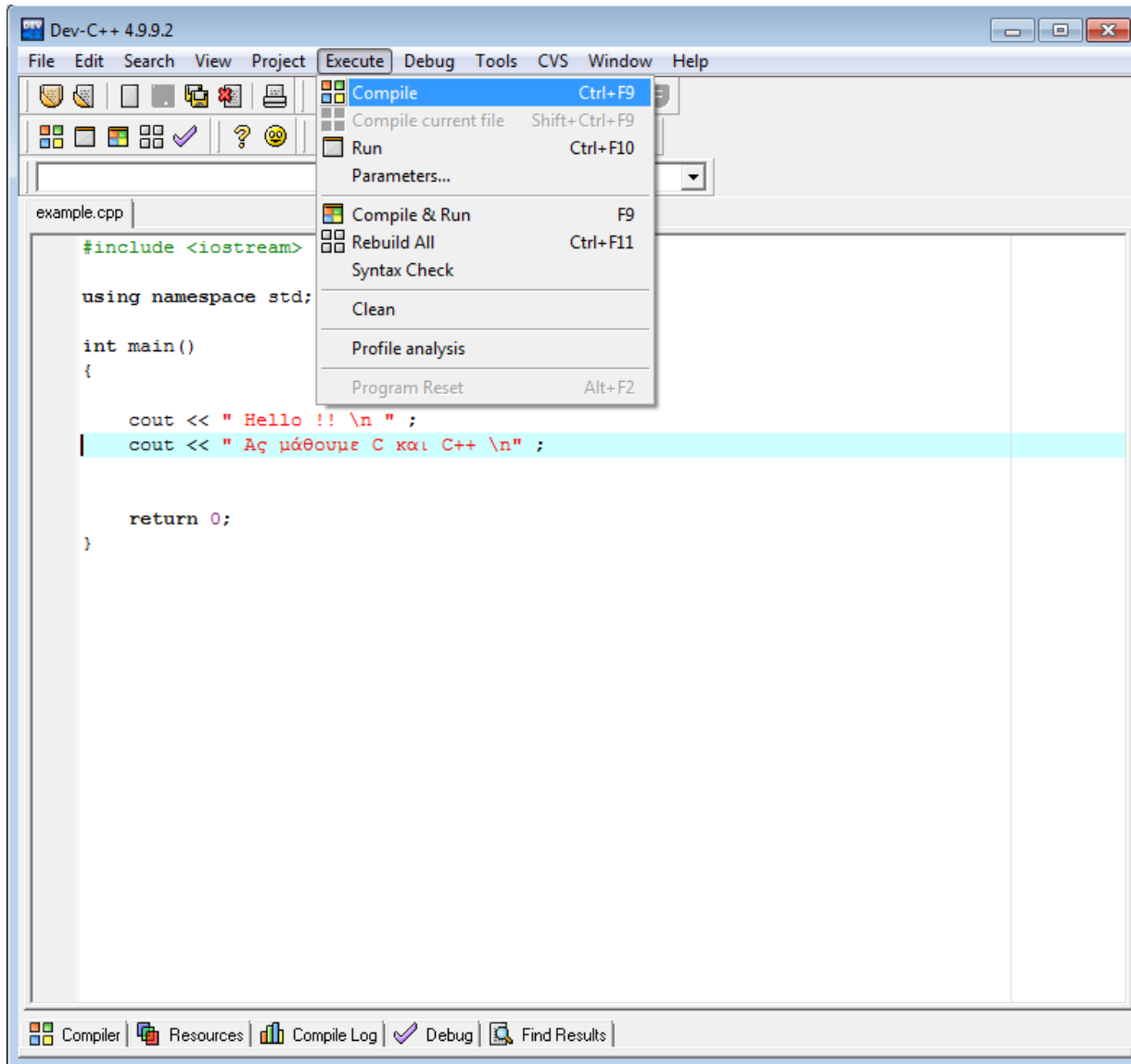
int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;
}
```

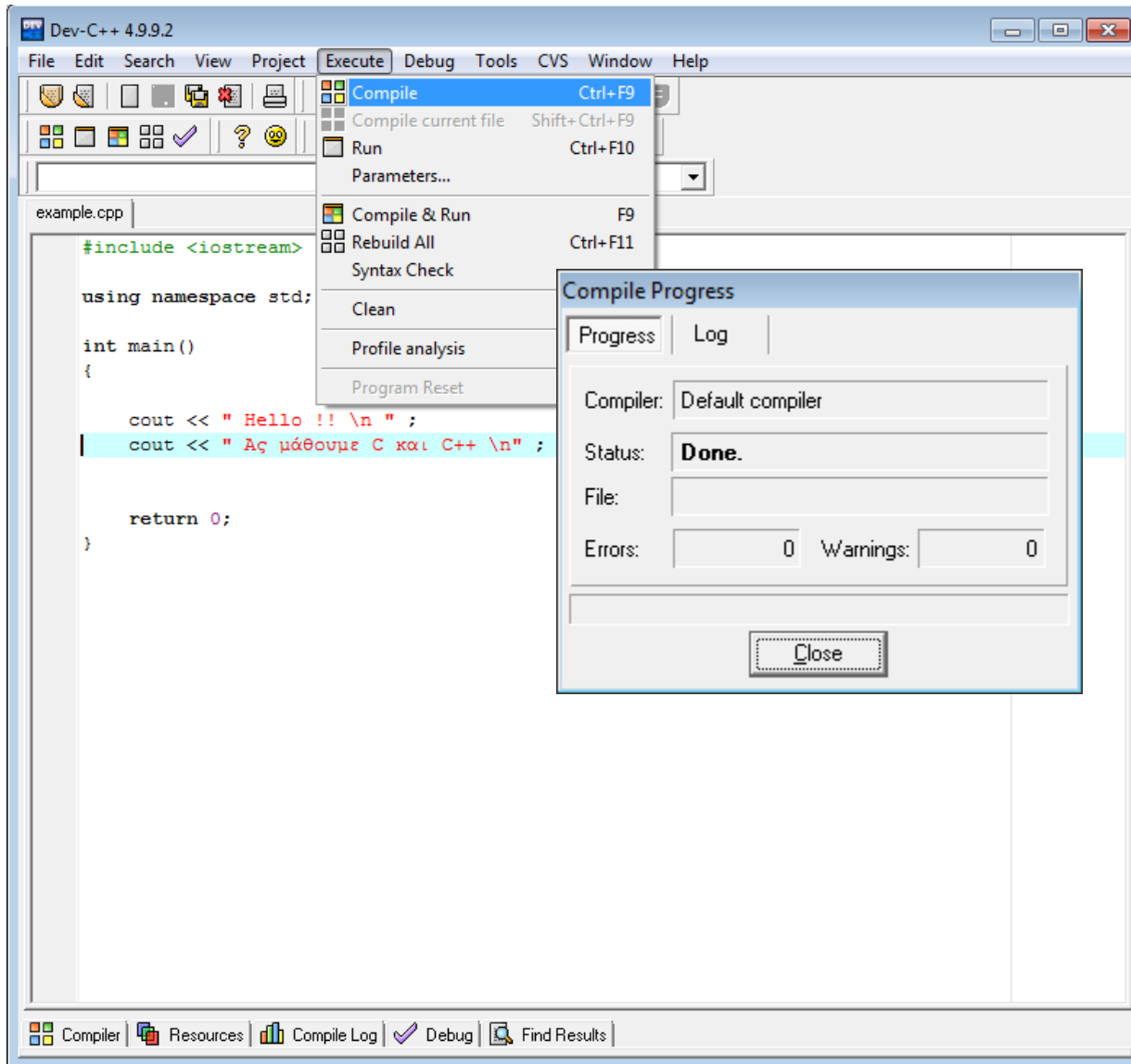
The second line of the cout statement is highlighted in light blue. The status bar at the bottom shows icons for Compiler, Resources, Compile Log, Debug, and Find Results.

# Στην πράξη;



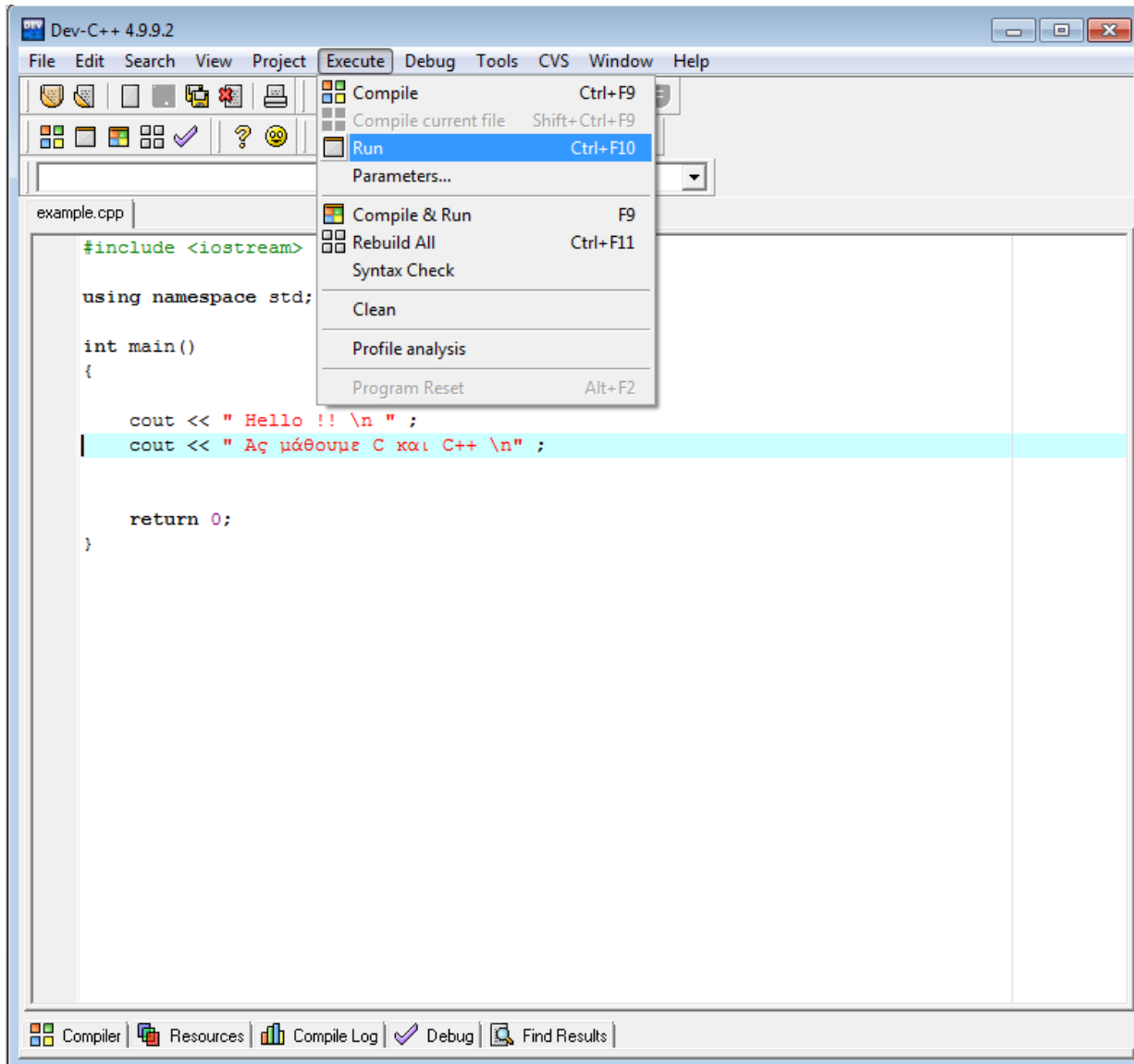
- Μετάφραση (compile)
- Προσοχή στα συντακτικά λάθη

# Στην πράξη;



- Μετάφραση (compile)
- Προσοχή στα συντακτικά λάθη

# Στην πράξη;



- Τρέξιμο

# Στην πράξη;

A screenshot of the Dev-C++ 4.9.2 IDE. The window title is "Dev-C++ 4.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. Below the toolbar is a dropdown menu. The main editor area shows a C++ program named "example.cpp" with the following code:

```
#include <iostream>

using namespace std;

int main()
{
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;
}
```

The second line of the main function is highlighted in light blue. The status bar at the bottom shows "Compiler", "Resources", "Compile Log", "Debug", and "Find Results".

- Τρέξιμο
- Αποτέλεσμα δεν βλέπουμε!!
- Γιατί;

# Στην πράξη;

A screenshot of the Dev-C++ 4.9.2 IDE. The window title is "Dev-C++ 4.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. The main editor area shows a C++ program named "example.cpp" with the following code:

```
#include <iostream>

using namespace std;

int main()
{
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;
}
```

The second line of the main function is highlighted in light blue. The status bar at the bottom shows "Compiler", "Resources", "Compile Log", "Debug", and "Find Results".

- Τρέξιμο
- Αποτέλεσμα δεν βλέπουμε!!
- Γιατί;
- Εκτυπώνει σε ένα τερματικό παράθυρο (νέο παράθυρο) τα κείμενα και στο τέλος το παράθυρο κλείνει μόνο του

# Στην πράξη;

A screenshot of the Dev-C++ 4.9.9.2 IDE. The window title is "Dev-C++ 4.9.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, editing, and debugging. The main editor area shows a C++ program named "example.cpp" with the following code:

```
#include <iostream>

using namespace std;

int main()
{
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;
    system("PAUSE");
    return 0;
}
```

The line `system("PAUSE");` is highlighted in light blue. The status bar at the bottom shows "Compiler", "Resources", "Compile Log", "Debug", and "Find Results".

- Τρέξιμο
- Προσθέτουμε την εντολή:  
`system("PAUSE");`
- ακριβώς πριν το τέλος
  
- Θα αναγκάσει το τερματικό παράθυρο να μείνει ανοικτό μέχρι να πατήσουμε κάποιο πλήκτρο



# Στην πράξη;



The screenshot shows the Dev-C++ 4.9.9.2 IDE. The main window displays a C++ program named 'example.cpp' with the following code:

```
#include <iostream>

using namespace std;

int main()
{
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    system("PAUSE");
    return 0;
}
```

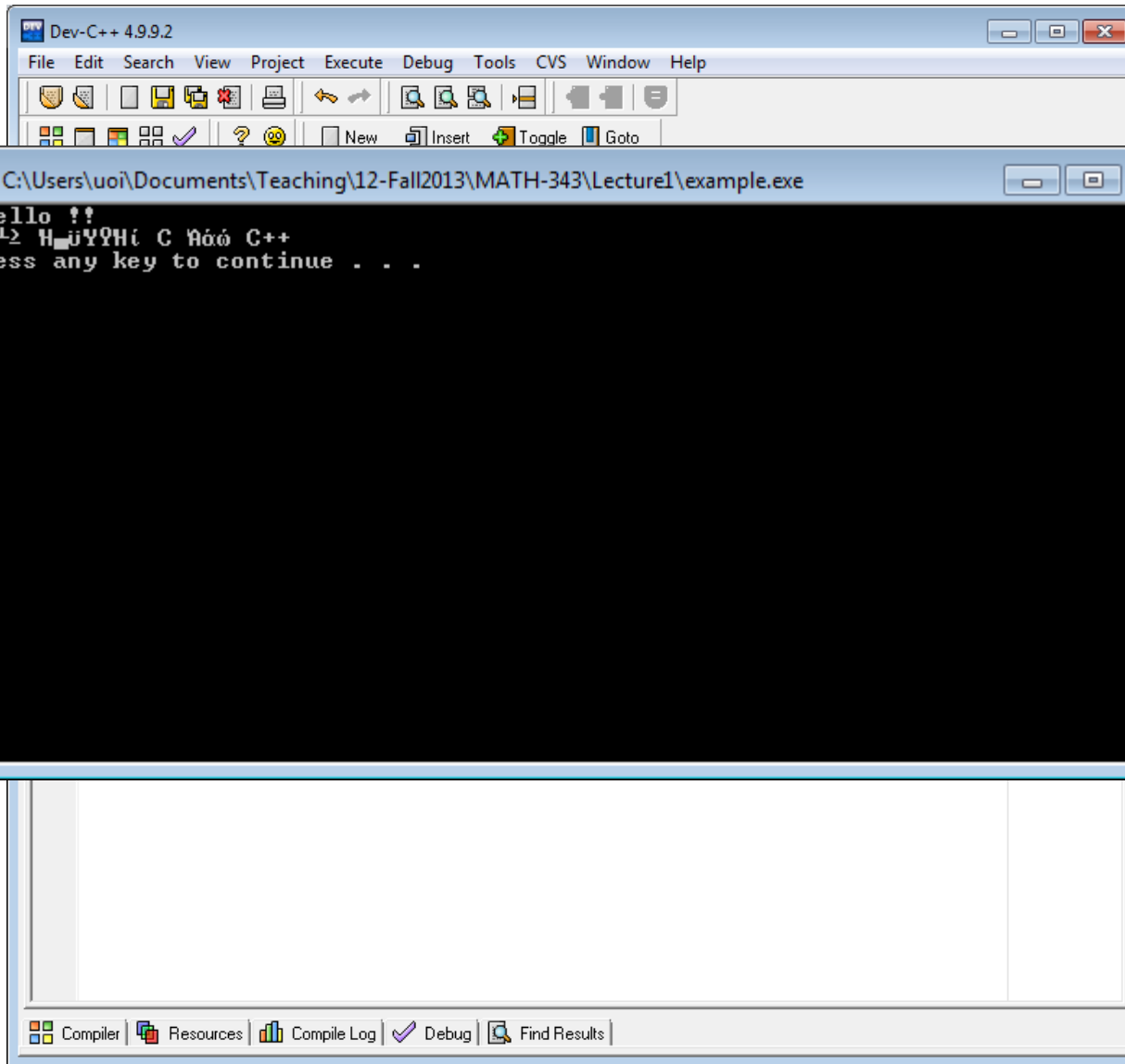
The 'Execute' menu is open, showing options: Compile (Ctrl+F9), Compile current file (Shift+Ctrl+F9), Run (Ctrl+F10), Parameters..., Compile & Run (F9), Rebuild All (Ctrl+F11), Syntax Check, Clean, Profile analysis, and Program Reset (Alt+F2). The 'system("PAUSE");' line in the code is highlighted in blue.

- ΣΗΜΑΝΤΙΚΟ!!
  - Για να τρέξουμε πάλι το πρόγραμμα πρέπει να δημιουργήσουμε νέο εκτελέσιμο.
- ⇒ Πρέπει πάλι να το μεταφράσουμε

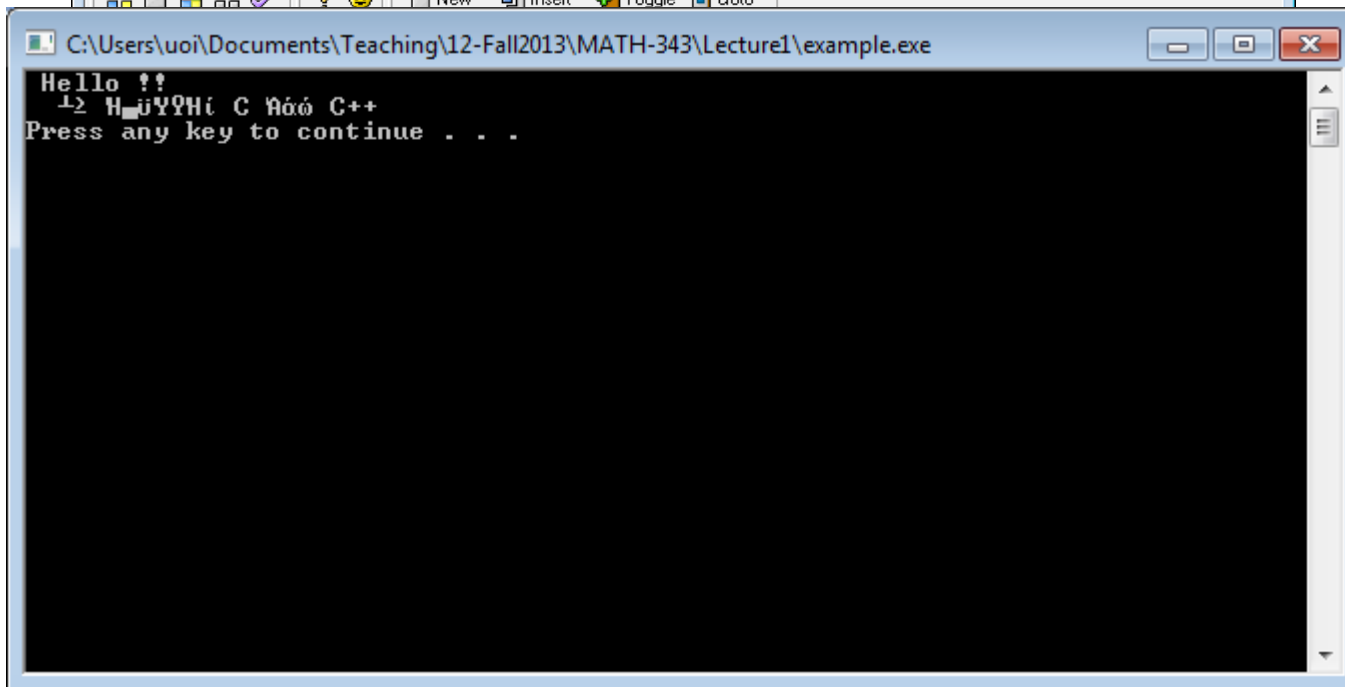
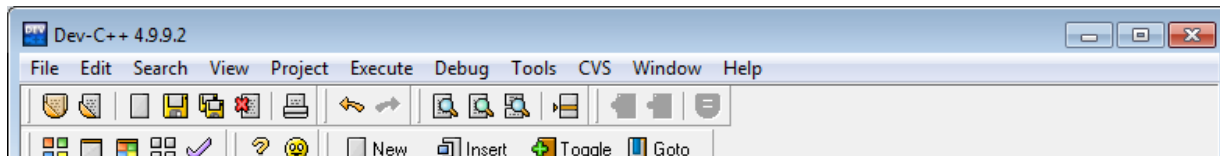
# Στην πράξη;



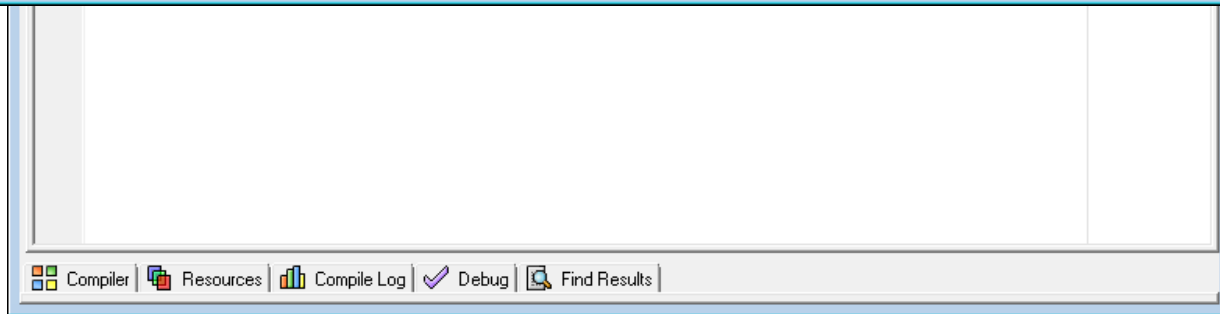
- Τρέξιμο (Run)
- Πατώντας οποιοδήποτε πλήκτρο κλείνει το παράθυρο



# Στην πράξη;



- Αν βγαίνουν κινέζικα τα ελληνικά τότε πρέπει να προσθέσουμε την εντολή `system("chcp 1253");`
- που βρίσκει την ελληνική κωδικοποίηση
- Σε ορισμένα συστήματα μπορεί να βγαίνει σωστά



# Στην πράξη;

A screenshot of the Dev-C++ 4.9.9.2 IDE. The window title is "Dev-C++ 4.9.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, execution, and debugging. The main editor area shows a C++ program named "example.cpp". The code is as follows:

```
#include <iostream>

using namespace std;

int main()
{
    system("chcp 1253");
    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    system("PAUSE");
    return 0;
}
```

The line `system("chcp 1253");` is highlighted in light blue. The status bar at the bottom shows "Compiler", "Resources", "Compile Log", "Debug", and "Find Results".

- Αν βγαίνουν κινέζικα τα ελληνικά τότε πρέπει να προσθέσουμε την εντολή `system("chcp 1253");`
- που βρίσκει την ελληνική κωδικοποίηση
- Σε ορισμένα συστήματα μπορεί να βγαίνει σωστά

# Στην πράξη;

A screenshot of the Dev-C++ 4.9.9.2 IDE. The window title is "Dev-C++ 4.9.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations and execution. The main editor area shows a file named "example.cpp" with the following code:

```
#include <iostream>

using namespace std;

int main()
{
    system("chcp 1253");
    cout << " Hello !! \n " ;
}
```

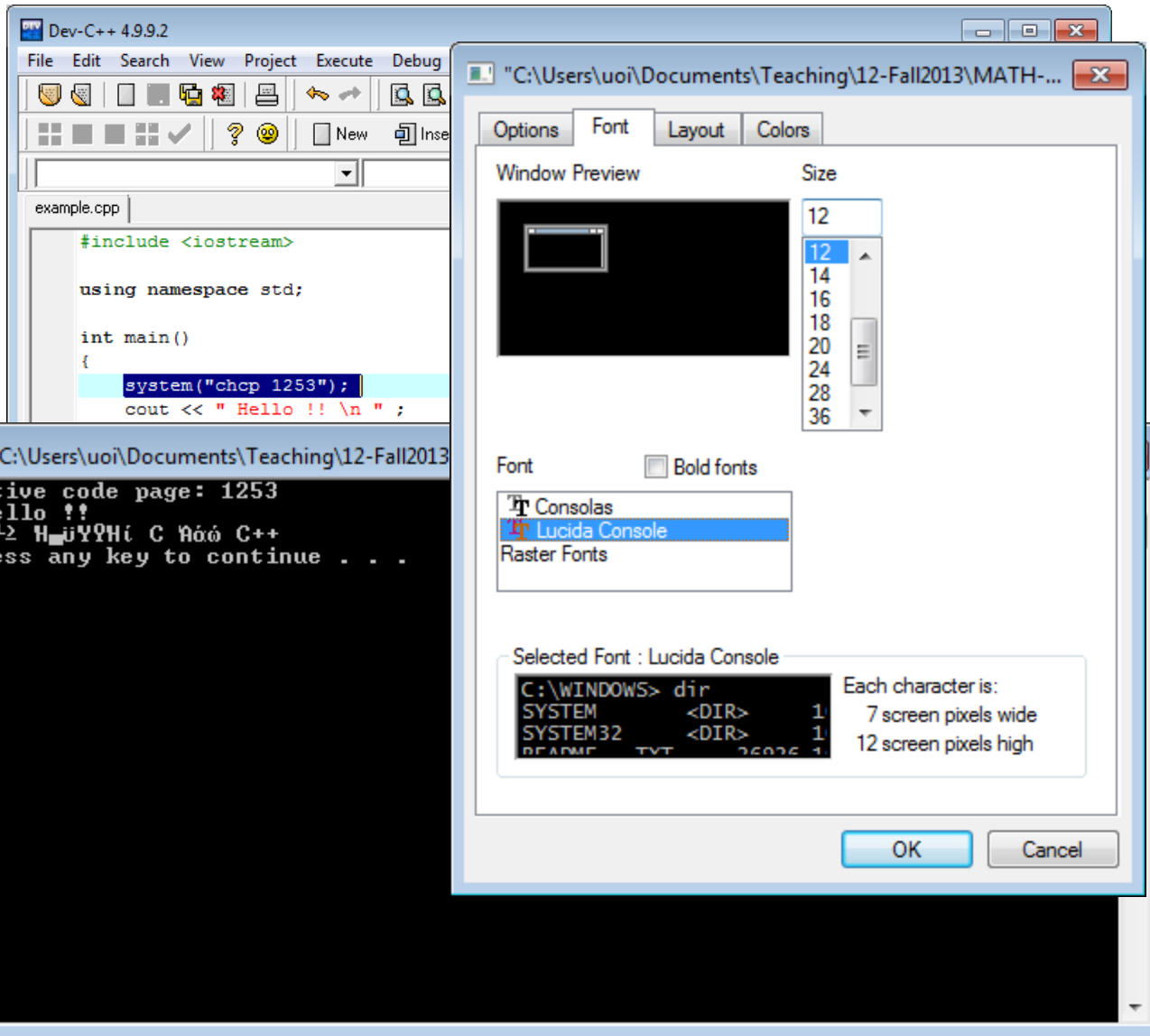
The line `system("chcp 1253");` is highlighted in light blue.

- Ξανά:
  1. compile
  2. run

A screenshot of a Windows command prompt window. The title bar shows the file path: "C:\Users\σοι\Documents\Teaching\12-Fall2013\MATH-343\Lecture1\example.exe". The window content shows the output of the program:

```
Active code page: 1253
Hello !!
  ↳ Ημῶν γὰρ ἡ C γλώσσα C++
Press any key to continue . . .
```

# Στην πράξη;



- Ξανά:
  1. compile
  2. run
- Αλλάζουμε την γραμματοσειρά ΤΟΥ Τερματικού:
  - Κλικ στο εικονίδιο του παραθύρου πάνω αριστερά
  - Ιδιότητες
  - Γραμματοσειρά Lucida

# Στην πράξη;

A screenshot of the Dev-C++ 4.9.9.2 IDE. The window title is "Dev-C++ 4.9.9.2". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. The main editor area shows a file named "example.cpp" with the following code:

```
#include <iostream>

using namespace std;

int main()
{
    system("chcp 1253");
    cout << " Hello !! \n " ;
}
```

- Ξανά:

1. compile
2. run

- Αλλάζουμε την γραμματοσειρά ΤΟΥ

- τερματικού:

- Κλικ στο εικονίδιο του παραθύρου πάνω αριστερά
- Ιδιότητες
- Γραμματοσειρά Lucida

A screenshot of a Windows command prompt window. The title bar shows the path "C:\Users\σοι\Documents\Teaching\12-Fall2013\MATH-343\Lecture1\example.exe". The window content shows the output of the program:

```
Active code page: 1253
Hello !!
Ας μάθουμε C και C++
Press any key to continue . . . _
```

# Σύνοψη του παραδείγματος

```
#include <iostream>

using namespace std;

int main()
{

    cout << " Hello !! \n " ;
    cout << " Ας μάθουμε C και C++ \n" ;

    return 0;

}
```

- Ένα πρόγραμμα αποτελείται από την κύρια συνάρτηση (`main`)
- Εκτέλεση προγράμματος  $\Rightarrow$  Εκτέλεση της `main`
- Κυρίως σώμα (εντολές) της `main`: περικλείεται από `{ }`
- Στη θεωρία δεν θα ασχοληθούμε με τις πολύ σημαντικές λεπτομέρειες για το τρέξιμο (εργαστήρια)



Ενότητα 2

# **ΔΥΑΔΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ**

# Συστήματα Αρίθμησης

- Ένας δεκαδικός αριθμός αποτελείται από μία ακολουθία δεκαδικών ψηφίων και ίσως από μία υποδιαστολή.
- Οποιοσδήποτε αριθμός εκφράζεται :  $Number = \sum_{i=-k}^n d_i \times 10^i$
- Π.χ.  $3347.4 = 3 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 + 4 \times 10^{-1}$
- Το σύστημα αυτό ονομάζεται δεκαδικό, λόγω του ότι ως εκθετική βάση έχει επιλεγεί το 10

# Συστήματα Αρίθμησης

Π.χ.  $3347.4 = 3 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 + 4 \times 10^{-1}$

$$Number = \sum_{i=-k}^n d_i \times 10^i$$

*Τι εκφράζουν τα  $n$   
και  $k$  στο πάνω  
παράδειγμα;*

- Αντίστοιχα αριθμητικά συστήματα με βάση το 2, 8 και 16 είναι το *δυναδικό, οκταδικό, δεκαεξαδικό*
- Ένα αριθμητικό σύστημα με βάση το  $k$  απαιτεί  $k$  διαφορετικά σύμβολα για την αναπαράσταση των ψηφίων

# Θεσιακά Συστήματα

Π.χ.  $3347.4 = 3 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 + 4 \times 10^{-1}$

Ως *βάση* ή *ρίζα* ενός θεσιακού συστήματος ορίζεται το πλήθος των διαφορετικών ψηφίων:

- Δεκαδικό: **0 1 2 3 4 5 6 7 8 9**
- Δυαδικό: **0 1**
- Οκταδικό : **0 1 2 3 4 5 6 7**
- Δεκαεξαδικό : **0 1 2 3 4 5 6 7 8 9 A B C D E F**

# Παραδείγματα

---

$$(1673.42)_{10} = 1 \times 10^3 + 6 \times 10^2 + 7 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 2 \times 10^{-2}$$

$$(100110)_2 = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (38)_{10}$$

$$(372)_8 = 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = (250)_{10}$$

$$(A34F)_{16} = A \times 16^3 + 3 \times 16^2 + 4 \times 16^1 + F \times 16^0 = (41807)_{10}$$

# Μετατροπή από το δυαδικό στο δεκαδικό σύστημα

128	64	32	16	8	4	2	1
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

Το περισσότερο σημαντικό bit

Το λιγότερο σημαντικό bit

1 1 1 1 0 0 1 1

$1 \times 128$   $1 \times 64$   $1 \times 32$   $1 \times 16$   $0 \times 8$   $0 \times 4$   $1 \times 2$   $1 \times 1$

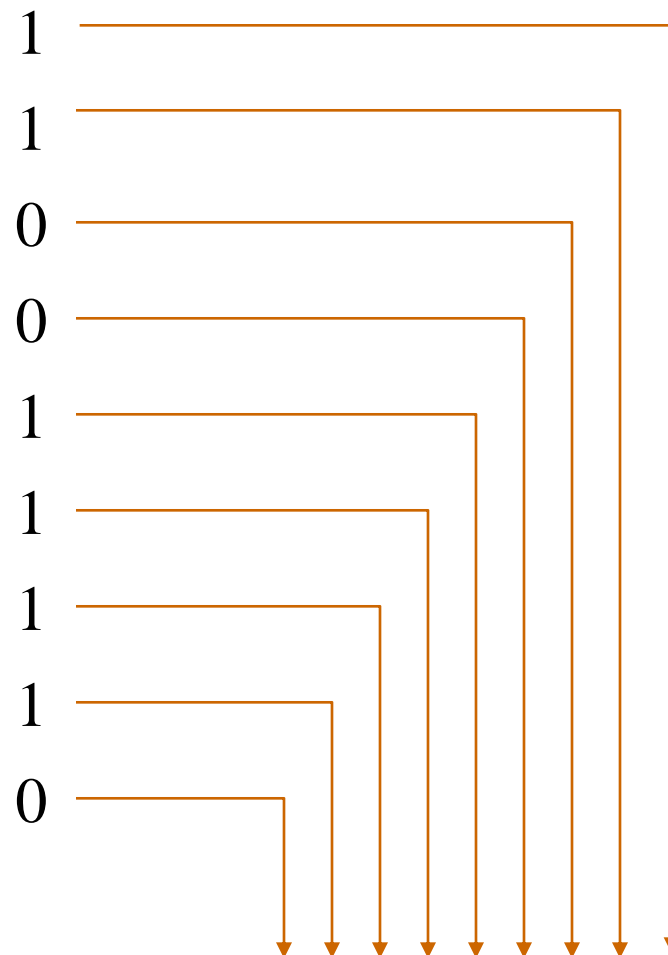
$128 + 64 + 32 + 16 + 0 + 0 + 2 + 1 =$

$= 243$  (δεκαδικό)

# Μετατροπή δεκαδικού σε δυαδικό

- Από δεκαδικό σε δυαδικό (Διαδοχικές διαιρέσεις με το 2) :

2 4 3  
1 2 1  
6 0  
3 0  
1 5  
7  
3  
1  
0



*Εξετάζοντας  
το υπόλοιπο  
της διαίρεσης*

$$011110011 = (243)_{10}$$

# Μετατροπή δεκαδικού σε οκταδικό

- Από δεκαδικό σε οκταδικό (Διαδοχικές διαιρέσεις με το 8) :

7 6 5 3

9 5 6

1 1 9

1 4

1

5

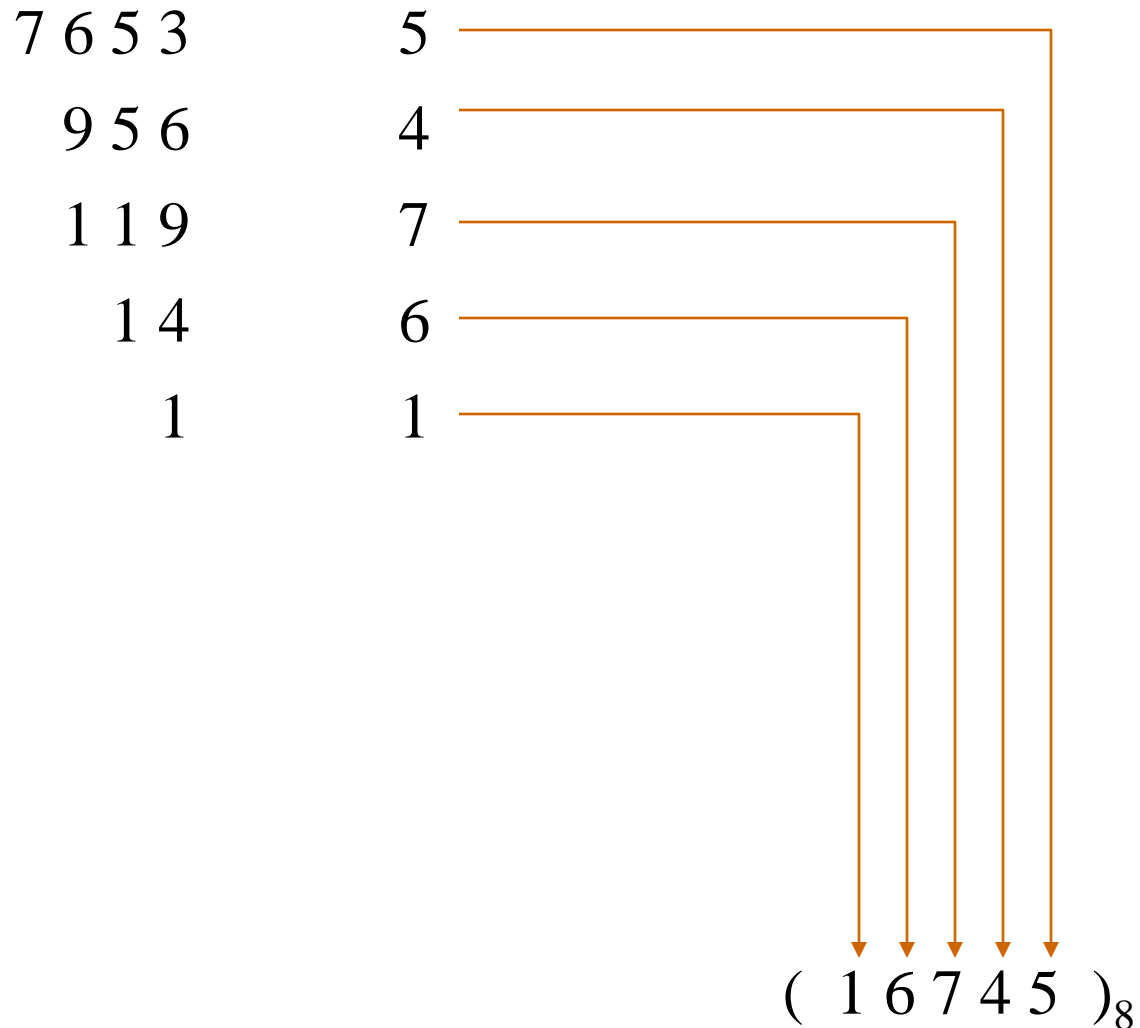
4

7

6

1

( 1 6 7 4 5 )<sub>8</sub>





# Μετατροπή δεκαδικού σε οκταδικό

- Ευκολότερα με πίνακες μετατροπής από δυαδικό σε οκταδικό

Δυαδικό	Οκταδικό
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

$$(411)_{10} = (1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1)_2$$

$\underbrace{\hspace{1.5em}} \quad \underbrace{\hspace{1.5em}} \quad \underbrace{\hspace{1.5em}}$

**6      3      3**

$$(411)_{10} = (633)_8$$

# “Συντομογραφίες” δυαδικών αριθμών

- Δεκαεξαδικό σύστημα
  - Ομαδοποίηση ανά 4 bits

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

# Παράδειγμα στο δεκαεξαδικό σύστημα

- Παράδειγμα: 1100100110010100

1100 1001 1001 0100

$$C \quad 9 \quad 9 \quad 4 = (C994)_{16}$$

- Παράδειγμα: 10000101011110

0010 0001 0101 1110

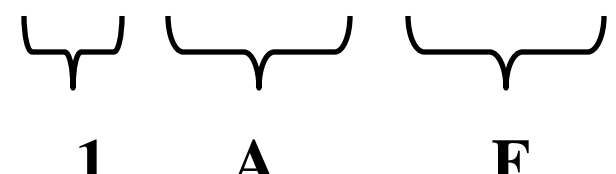
$$2 \quad 1 \quad 5 \quad E = (215E)_{16}$$

- Συμπλήρωση με 0 στα αριστερά
- Δεν αλλάζει τον αριθμό, όπως ακριβώς και στο δεκαδικό σύστημα

# Παράδειγμα στο δεκαεξαδικό σύστημα

- Από δεκαδικό σε δεκαεξαδικό :

$$(431)_{10} = (1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1)_2$$



**1      A      F**

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

# Αριθμοί σε διάφορες βάσεις

Δεκαδικό	Δυαδικό	Οκταδικό	Δεκαεξαδικό
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

# Φυσικοί αριθμοί (χωρίς πρόσημο)

- Άμεση αντιστοιχία
- Με  $n$  bits περιγράφονται:  
**Οι φυσικοί αριθμοί από  
 $0$  έως και  $2^n - 1$**

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
...	...

# Καλή Μελέτη

- **Βιβλιογραφία**

[1] W. Savitch, Πλήρης C++, Εκδόσεις Τζιόλα, 2011

[2] H. Deitel and P. Deitel, C++ Προγραμματισμός 6η Εκδοση, Εκδόσεις Μ. Γκιούρδας, 2013

## Ύλη βιβλιογραφίας

[1]: 1.1, Παράρτημα 3

[2]: Κεφ. 1, Παραρτήματα Β, Δ