

# 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

# 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 Μθηματικών  
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, διάθωρα σταθμάντων, 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. Διαφορικά (ζωή προσδιοριστών και κανόνες εμφάνισης και ορατότητας. Χρήση αρχείων. Δομές, εγγραφές, λίστες με αλφριθμούς και διαγράμματα ροής προγραμμάτων. Εφαρμογές σε προβλήματα αναζήτησης, ταξινόμησης και μαθηματικών προβλημάτων.

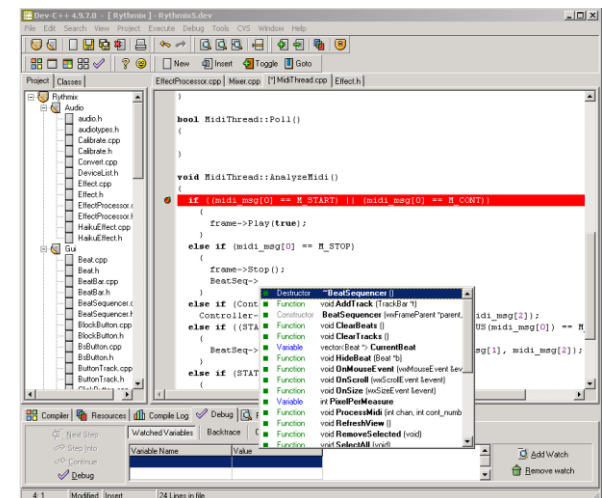
Στο μάθημα περιλαμβάνονται εργαστηριακές ασκήσεις στις οποίες η συμμετοχή είναι υποχρεωτική.

## 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. Χατζηγιαννάκης, Η γλώσσα C++ σε βάθος, Εκδόσεις Κλειδάριθμος, 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	

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

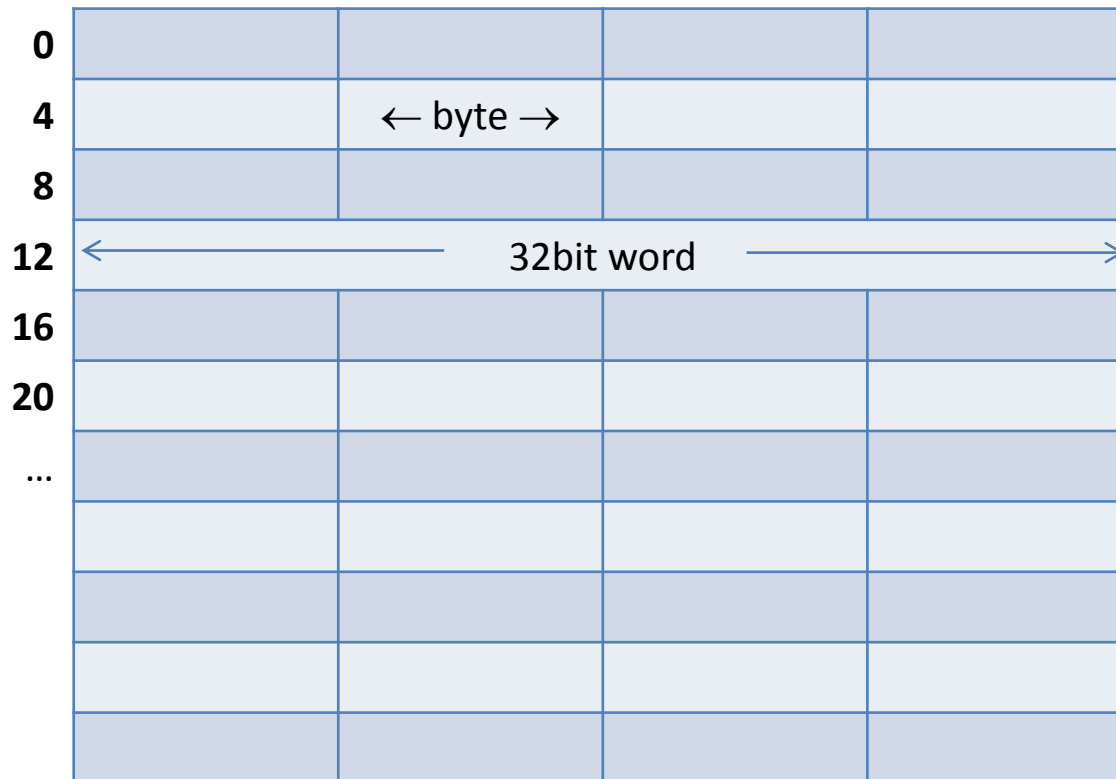
Εβδομάδα	Θέματα	Υψη βιβλιογραφίας
Πα, 9 Οκτωβρίου	Εισαγωγικά μαθήματος & Δυαδική αναπαράσταση	[1]: 1.1, Παράρτημα 3 [2]: Κεφ. 1, Β, Δ
Πα, 16 Οκτωβρίου	Είσοδος/Εξοδος δεδομένων, τύποι δεδομένων & μεταβλητών	[1]: 1.2, 1.3, 1.4, 1.5, Παράρτημα 1 [2]: Κεφ. 2, Γ
Δε, 17 Οκτ	1 <sup>ο</sup> Εργαστήριο	
Πα, 23 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	

Ενότητα 3

# **ΜΕΤΑΒΛΗΤΕΣ, ΕΚΦΡΑΣΕΙΣ & ΕΝΤΟΛΕΣ ΕΚΧΩΡΗΣΗΣ**

# Η μνήμη του υπολογιστή

- Η μνήμη είναι σαν ένας πίνακας και μπορούμε να προσπελάσουμε ένα στοιχείο της μνήμης με έναν αριθμό.



- Στο παραπάνω παράδειγμα: 32-bit αρχιτεκτονική.

# Μεταβλητές

- Αντί να θυμόμαστε αριθμούς και διευθύνσεις, χρησιμοποιούμε μεταβλητές σε ένα πρόγραμμα

```
#include <iostream>

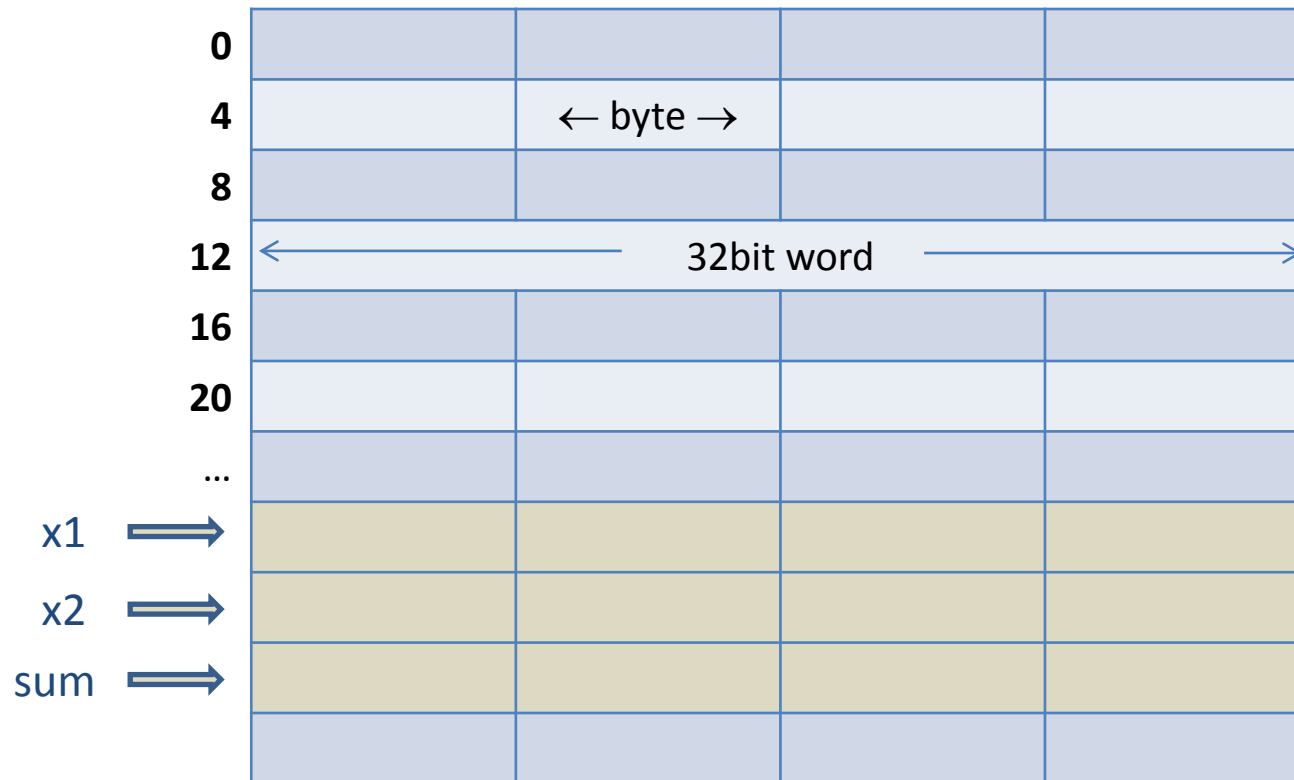
int main()
{
    int x1, x2, sum;

    return 0;
}
```

- Λέμε στον υπολογιστή να δεσμεύσει 3 θέσεις μνήμης για να αποθηκεύσουμε ακέραιους αριθμούς
- Αυτές οι 3 θέσεις μνήμης ονομάζονται x1, x2, και sum
- Οι διευθύνσεις αναφέρονται με τα παραπάνω ονόματα



# Μεταβλητές



- Ουσιαστικά τα ονόματα είναι συντομεύσεις για τις διευθύνσεις

# Μεταβλητές

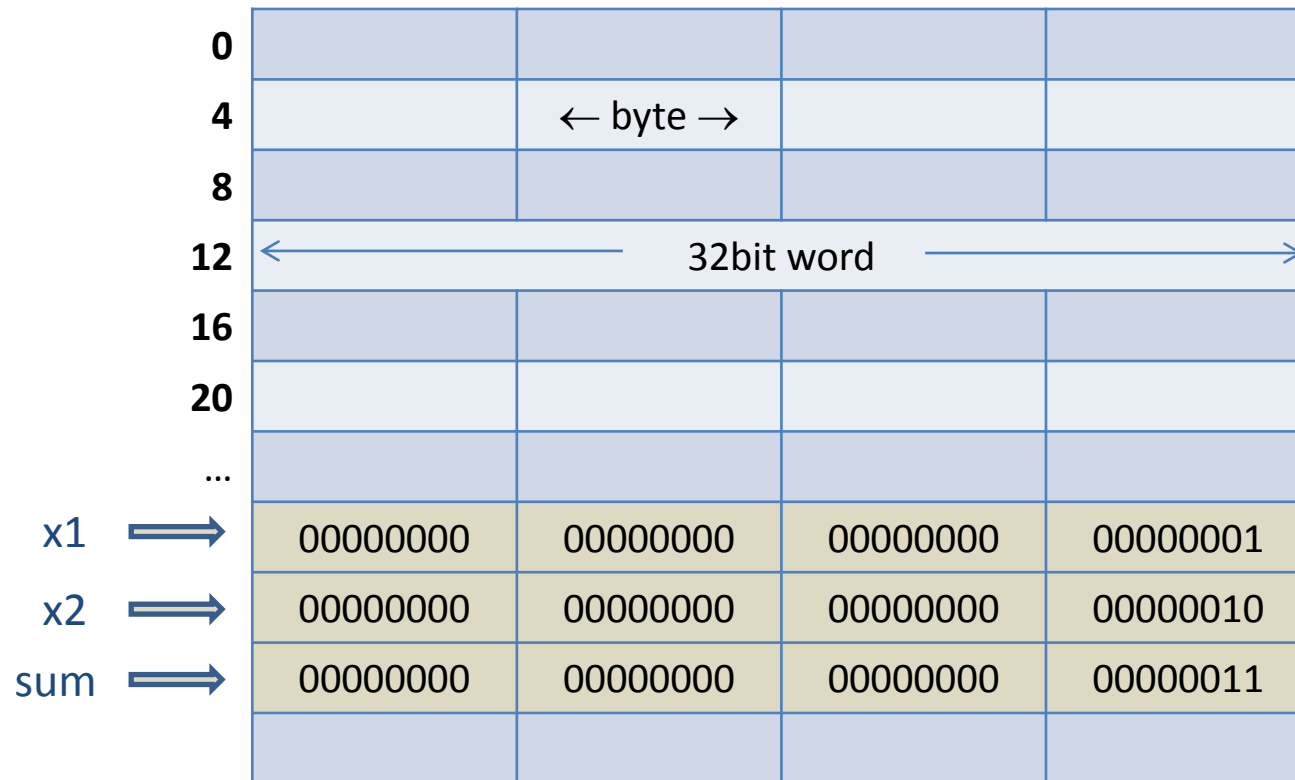
- Μετά την δέσμευση 3 θέσεων στην μνήμη μπορούμε να αποθηκεύσουμε τιμές στις αντίστοιχες θέσεις

```
#include <iostream>

int main()
{
    int x1, x2, sum;
    x1 = 1;
    x2 = 2;
    sum = x1 + x2;
    return 0;
}
```

- Αποθηκεύει την τιμή 1 στην θέση μνήμης x1
- Αποθηκεύει την τιμή 2 στην θέση μνήμης x2
- Αποθηκεύει την τιμή 3 στην θέση μνήμης sum

# Μεταβλητές



- Ουσιαστικά τα ονόματα είναι συντομεύσεις για τις διευθύνσεις

# Μεταβλητές στη C++

- Ονοματολογία μεταβλητών:

- Κάθε μεταβλητή πρέπει να αρχίζει με: γράμμα ή \_ (κάτω παύλα)
- Υπόλοιποι χαρακτήρες: γράμματα ή αριθμοί ή \_

Π.χ.:            `x`            `x1`            `x_1`            `_abc`            `data2`  
                  `12`            `3x`            `%count`

- Case sensitive γλώσσα

- Ξεχωρίζει τους πεζούς από τους κεφαλαίους χαρακτήρες
- 3 διαφορετικές μεταβλητές: `rate`    `RATE`    `Rate`

- Δεσμευμένες λέξεις

- Χρήση μόνο για τον προκαθορισμένο τους σκοπό
- Δεν χρησιμοποιούνται για ονοματολογία μεταβλητών (συναρτήσεων)

# Δεσμευμένες λέξεις στη C++

asm	do	inline	short	typeid
auto	double	int	signed	typename
bool	dynamic_cast	long	sizeof	union
break	else	mutable	static	unsigned
case	enum	namespace	static_cast	using
catch	explicit	new	struct	virtual
char	extern	operator	switch	void
class	false	private	template	volatile
const	float	protected	this	wchar_t
const_cast	for	public	throw	while
continue	friend	register	true	
default	goto	reinterpret_cast	try	
delete	if	return	typedef	

- Εκτός από τις λέξεις του πίνακα, δεσμευμένες λέξεις είναι οτιδήποτε ξεκινάει με `__`
  - Χρησιμοποιούνται σε τυποποιημένες βιβλιοθήκες
  - Π.χ. : `__count` `__i` `__x1`

Δεν πρέπει να δηλώνουμε τέτοιες μεταβλητές!!

# Δήλωση μεταβλητών

- Κάθε μεταβλητή έχει:

① όνομα

② τύπο

③ τιμή

Τύπος: είδος πληροφορίας που αποθηκεύουμε στην θέση μνήμης

## Βασικοί τύποι μεταβλητών

1. ακέραιοι: `int x;`

2. χαρακτήρες: `char x;`

3. κινητής υποδιαστολής (προσεγγιστική αναπαράσταση πραγματικών αριθμών): `double x;`

4. λογικές (δυναδικές): `bool x;` (true-αληθής ή false-ψευδής)

- Άλλοι τύποι μεταβλητών: σε λίγο... (αναπαράσταση αριθμών)

# Δήλωση μεταβλητών

- Παράδειγμα:

```
int x1, x2;  
double sum;
```

- Τελειώνουν με ;
  - Όταν υπάρχουν περισσότερες από μια μεταβλητές (<λίστα>) τότε χωρίζονται με ,
- 
- Κάθε μεταβλητή πρέπει να δηλωθεί πριν την πρώτη χρησιμοποίησή της.
    - Συνήθως πριν την χρησιμοποίησή της  
ή
    - στην αρχή κάποιου τμήματος του προγράμματος που ξεκινάει με {

```
int x, y, z;  
x = 1;  
y = 2;  
  
i = 0;  
int i; ΛΑΘΟΣ
```

# Δήλωση μεταβλητών - Σύνταξη

## Σύνταξη

*όνοματύπου ονομαμεταβλητης1, ονομαμεταβλητης2, ...;*

## Παράδειγμα

```
int count, numberOfDragons, numberOfTrolls;  
double distance;
```

- Κάθε μεταβλητή δηλώνεται πριν χρησιμοποιηθεί
- Η σύνταξη διαφοροποιείται όταν θέλουμε να δώσουμε αρχικές τιμές στην δήλωση μεταβλητών



# Απλοί τύποι μεταβλητών

Τύπος	Χρήση μνήμης	Περιοχή Τιμών
short (ή short int)	2 bytes	-32768 έως 32767
int	4 bytes	-2146473648 έως 2147483647
long (ή long int)	10 bytes	$\pm 9223372036854775807$
float	4 bytes	$10^{-38}$ έως $10^{38}$
double	8 bytes	$10^{-304}$ έως $10^{308}$
long double	10 bytes	$10^{-4932}$ έως $10^{4932}$
char	1 byte	Όλοι οι χαρακτήρες ASCII
bool	1 byte	false ή true

- Υπάρχουν και άλλοι τύποι μεταβλητών όπως
  - αλφαριθμητικοί χαρακτήρες: `string`
  - Πίνακες (arrays), κλάσεις...
- Για τους ακεραίους υπάρχει **μη-προσημασμένη** παραλλαγή:  
`unsigned short`, `unsigned int`, `unsigned long`

# Εντολές εκχώρησης

- Αλλαγή τιμής σε μια μεταβλητή

*ονομαμεταβλητης* = *έκφραση* ;

*έκφραση*:

μεταβλητή ή αριθμός ή σύνθετη έκφραση  
αποτελούμενη από μεταβλητές αριθμούς, τελεστές, συν/σεις

- Αποτίμηση της έκφρασης στο δεξί μέλος και στη συνέχεια θέτουμε την τιμή της έκφρασης στη τιμή της μεταβλητής

```
totalWeight = oneWeight * number ;
```

```
temperature = 37.5 ;
```

```
count = count + 2 ;
```

# Εντολές εκχώρησης

- Οι εντολές εκχώρησης είναι *εκφράσεις*
  - Επιστρέφουν την τιμή που αναθέτουμε στη μεταβλητή (αριστερό μέρος)  
 $n = (m = 2);$       ή       $n = m = 2;$
  - Η υποέκφραση  $m = 2$  αλλάζει την τιμή της μεταβλητής  $m$  σε 2 και επιστρέφει την τιμή 2 (που την αναθέτει τελικά και στην μεταβλητή  $n$ )
- Συνήθως δεν την χρησιμοποιούμε αλλά αντιλαμβανόμαστε καλύτερα σε κάποια κομμάτια κώδικα ενδεχόμενα λάθη

# Αρχικές τιμές

- Κάθε μεταβλητή πρέπει να έχει μια αρχική τιμή εκτός αν υποδηλώνεται έμμεσα

```
int x, y;  
y = x + 1;
```

- Η μεταβλητή  $x$  δεν έχει αρχικοποιηθεί και δεν γνωρίζουμε την τιμή που έχει (συνήθως θα έχει κάποιο τυχαίο αριθμό)

- Θα πρέπει να δώσουμε αρχική τιμή στην μεταβλητή  $x$ :

```
int x, y;  
x = 3;  
y = x + 1;
```

```
int x = 3, y;  
y = x + 1;
```

```
int x(3), y;  
y = x + 1;
```

Όλα τα παραπάνω είναι ισοδύναμα

# Αρχικές τιμές

## Απόδοση αρχικών τιμών σε δηλώσεις Σύνταξη

*όνοματύπου* *ονομαμεταβλητης1* = *εκφραση1*,  
*ονομαμεταβλητης2* = *εκφραση2*, ...;

## Παραδείγματα

```
int count = 0, limit = 10, x = 0;  
double distance = 999.99;
```

## Απόδοση αρχικών τιμών σε δηλώσεις Εναλλακτική Σύνταξη

*όνοματύπου* *ονομαμεταβλητης1* (*εκφραση1*),  
*ονομαμεταβλητης2* (*εκφραση2*), ...;

## Παραδείγματα

```
int count(0), limit(10), x(0);  
double distance(999.99);
```

# Ειδικές εντολές εκχώρησης

- Συντόμευση εντολών εκχώρησης  $a = a + 3;$   $a += 3;$

Η εντολή εκχώρησης,

*ονομαμεταβλητης* = *ονομαμεταβλητης* *τελεστης* *έκφραση* ;

μπορεί να γραφεί ισοδύναμα σε

*ονομαμεταβλητης* *τελεστης* = *έκφραση* ;

*τελεστης*: + , - , \* , / , % [βασικές αριθμητικές πράξεις]

- Παραδείγματα:

**d -= 4**            (**d = d - 4**)

**e \*= 5**            (**e = e \* 5**)

**f /= 3**            (**f = f / 3**)

**g %= 9**            (**g = g % 9**)

# Συμβατότητα εκχωρήσεων

Αποθήκευση τιμής ενός τύπου σε μια μεταβλητή άλλου τύπου

```
int myInt;  
myInt = 2.99; (myInt = 2)
```

- Από κινητής υποδιαστολής σε ακέραια:
  - κόβονται τα δεκαδικά ψηφία (συνήθως)

```
double myDouble;  
myDouble = 3; (myDouble = 3.0)
```

- Από ακέραια σε κινητής υποδιαστολής :
  - προστίθεται ".0" (πάντα)

# Συμβατότητα εκχωρήσεων

## Αποθήκευση τιμής ενός τύπου σε μια μεταβλητή άλλου τύπου

```
char myChar;  
myChar = 64; (myChar = 'Z')
```

```
int myInt;  
myInt = 'Z'; (myInt=64)
```

- Από **char** σε **int** και από **int** σε **char**:
  - αντιστοίχιση από τον πίνακα ASCII

```
bool myBool;  
myBool = 2; (myBool = true)
```

```
int myInt;  
myInt = false; (myInt=0)
```

- Από **bool** σε **int** και από **int** σε **bool** :
  - bool=int: μη-μηδενικός ακέραιος ως true, αλλιώς false
  - int=bool: Η τιμή true αποθηκεύεται ως 1 και η false ως 0



# Σταθερές

- Σταθερές τύπου `int`:

- δεν περιέχουν υποδιαστολή `int x = 3;`

- Σταθερές τύπου `double`:

- γίνεται και με συμβολογραφία κινητής υποδιαστολής:

- `double x = 3670000, y=3.67e3, z= 5.89e-6;`  
(3670 = 3.67\*10<sup>3</sup>) (0.00000589 = 5.89\*10<sup>-6</sup>)

- Σταθερές τύπου `char`:

- τοποθετώντας τον χαρακτήρα σε μονά εισαγωγικά `'`:

- `char symbol = 'Z';`

- οι αλφαριθμητικές σταθερές (strings) με διπλά εισαγωγικά `"`:

- `cout << " Hello !! " ;`

- Είναι διαφορετικό το `'Z'` από το `"Z" !!!`

- Σταθερές τύπου `bool`:

- έχει δύο μόνο τιμές `true` ή `false`: `bool x = true, y=false;`

# Πίνακας ASCII

Αριθμ	Χαρακτ	Αριθμ	Χαρακτ	Αριθμ	Χαρακτ	Αριθμ	Χαρακτ
32		56	8	80	P	104	h
33	!	57	9	81	Q	105	i
34	"	58	:	82	R	106	j
35	#	59	;	83	S	107	k
36	\$	60	<	84	T	108	l
37	%	61	=	85	U	109	m
38	&	62	>	86	V	110	n
39	.	63	?	87	W	111	o
40	(	64	@	88	X	112	p
41	)	65	A	89	Y	113	q
42	*	66	B	90	Z	114	r
43	+	67	C	91	[	115	s
44	,	68	D	92	\	116	t
45	-	69	E	93	]	117	u
46	.	70	F	94	^	118	v
47	/	71	G	95	_	119	w
48	0	72	H	96	`	120	x
49	1	73	I	97	a	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	c	123	{
52	4	76	L	100	d	124	
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	~
55	7	79	O	103	g		

# Ονομασία σταθερών (δεν αλλάζουν)

## Σύνταξη

```
const όνοματύπου ονομαμεταβλητης = Σταθερα;
```

## Παραδείγματα

```
const int MAX_TRIES = 3;  
const double PI = 3.14159;
```

- Οι δηλωμένες σταθερές (const) δεν αλλάζουν κατά τη διάρκεια του προγράμματος
- Συνήθως οι σταθερές μεταβλητές γράφονται με ΚΕΦΑΛΑΙΑ

# Παράδειγμα σταθερών

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

int main()
{
    const double RATE = 6.9;
    double deposit;

    cout << "Δώστε το ποσό κατάθεσης " ;
    cin >> deposit;

    double newBalance;
    newBalance = deposit + deposit*(RATE/100);
    cout << "Σε ένα χρόνο το ποσό θα γίνει " << newBalance;

    return 0;
}
```

- Τι γίνεται αν αλλάξει το επιτόκιο σε 5.5% ;

Δώστε το ποσό κατάθεσης 100  
Σε ένα χρόνο το ποσό θα γίνει 106.9

# Αριθμητικοί τελεστές

- Σχηματισμός εκφράσεων από
  - μεταβλητές, σταθερές, **τελεστές**
- Αριθμητικοί τελεστές μεταξύ
  - `int` και `double` τύπους

$$z = x \text{ **τελεστής** } y$$

- Τύπος παραγόμενης τιμής:
  - Αν `x` και `y` `int`  $\Rightarrow$  `z` `int`
  - Αν `x` ή `y` `double`  $\Rightarrow$  `z` `double`

Τελεστής	Περιγραφή
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
%	Υπόλοιπο διαίρεσης

Παράδειγμα:

```
total = baseamount + increase
```

η τιμή της `total` εξαρτάται από τους τύπους των τιμών `baseamount`, `increase`

# Προτεραιότητες τελεστών

Ο τρόπος υπολογισμού μιας έκφρασης εξαρτάται από την προτεραιότητα των τελεστών:

**1 παρενθέσεις: ()**

Υπολογίζονται πρώτα, από τα αριστερά προς τα δεξιά. Εάν υπάρχουν ένθετες υπολογίζονται πρώτα οι εσωτερικές

**2 πολλαπλασιασμός, διαίρεση και υπόλοιπο: \*, /, ή %**

Υπολογίζονται δεύτερα από αριστερά προς τα δεξιά.

**3 πρόσθεση, αφαίρεση: + ή -**

Εάν υπάρχουν πολλοί, υπολογίζονται από τα αριστερά προς τα δεξιά.

**4 εκχώρησης: =, +=, -=, \*=, /=, %=**

Από δεξιά προς τα αριστερά.

⇒ Στην πράξη: να χρησιμοποιείται σχεδόν πάντα ()

# Παραδείγματα προτεραιότητας

$$x=y=z \quad \Leftrightarrow \quad x=(y=z)$$

$$x+y+z \quad \Leftrightarrow \quad (x+y)+z$$

$$m = (a+b+c+d)/4 \quad \Leftrightarrow \quad m = \frac{a + b + c + d}{4}$$

$$m = a+b+c+d/4 \quad \Leftrightarrow \quad m = a + b + c + \frac{d}{4}$$

$$y = a * x * x + b * x + c \quad \Leftrightarrow \quad y = ax^2 + bx + c$$

# Ακέραια διαίρεση

Έκφραση	Αποτέλεσμα
<code>total = 10 / 3</code>	<code>total = 3</code> (απορρίπτεται το δεκαδικό μέρος)
<code>total = 10 / 3.0</code>	<code>total = 3.33</code>
<code>total = 10.0 / 3</code>	<code>total = 3.33</code>
<code>sum = 17 / 5</code>	<code>sum = 3</code> (απορρίπτεται το δεκαδικό μέρος)

- Αν χρησιμοποιούμε ακεραίους σε έναν τελεστή και θέλουμε να εκτυπώσουμε και το υπόλοιπο  $\Rightarrow$  %

```
cout << "Το 17 δια 5 ισούται με " << 17/5 << "\n";  
cout << "και υπόλοιπο " << 17%5 ;
```

```
Το 17 δια 5 ισούται με 3  
και υπόλοιπο 2
```



# Επισήμανση στην ακέραια διαίρεση

- Έστω ότι ένας μηχανικός αυτοκινητόδρομων σας ζητάει ένα πρόγραμμα που υπολογίζει το συνολικό κόστος για ένα μήκος λεωφόρου με τιμή χρέωσης 5000€ / km

```
totalPrice = 5000 * (meters / 1000.0);
```

Για `meters = 14800`  $\Rightarrow$  `totalPrice = 74000 €`

- Αν όμως,

```
totalPrice = 5000 * (meters / 1000);
```

τότε για `meters = 14800`  $\Rightarrow$  `totalPrice = 70000 € (!!!)`

# Μετατροπή τύπων

- Τι γίνεται αν θέλουμε να κάνουμε διαίρεση μεταξύ τύπων int;
  - Το αποτέλεσμα της διαίρεσης θα είναι int διότι και οι δυο μεταβλητές είναι τύπου int

```
int sum, total;  
double avg;  
sum=9; total=4;  
avg = sum /total;
```

- Μετατροπή τύπου (type cast) int σε τύπου double

## Σύνταξη

```
static_cast<double>(εκφραση)
```

## Παραδείγματα

```
double ans = n / static_cast<double>(m);
```

```
double ans = static_cast<double>(n) / m;
```

```
ΛΑΘΟΣ: double ans = static_cast<double>(n/m);
```

```
int sum, total;  
double avg;  
sum=9; total=4;  
avg = sum /static_cast<double>(total);
```

# Μοναδιαίοι τελεστές αύξησης μείωσης

- Ο τελεστής ++ αυξάνει μία μεταβλητή κατά 1      `n++` ή `++n`
- Ο τελεστής -- μειώνει μία μεταβλητή κατά 1      `n--` ή `--n`
- Εκτελέσιμες εντολές

```
int n=1, m=4;
n++;
cout << "Η τιμή της n άλλαξε σε " << n << "\n";
m--;
cout << "Η τιμή της m άλλαξε σε " << m << "\n";
```

```
Η τιμή της n άλλαξε σε 2
Η τιμή της m άλλαξε σε 3
```

- Μέσα σε μαθηματικές εκφράσεις
  - Επιστρέφουν την τιμή και μετά αλλάζουν τιμή      `n++` ή `m--`
  - Αλλάζουν τιμή και μετά επιστρέφουν τιμή      `++n` ή `--m`

```
int n=2;
int x = 2*(n++);
cout << x;
```

4

```
int n=2;
int x = 2*(++n);
cout << x;
```

6

# Προτεραιότητες τελεστών

Ο τρόπος υπολογισμού μιας έκφρασης εξαρτάται από την προτεραιότητα των τελεστών:

**1 παρενθέσεις: ()**

Υπολογίζονται πρώτα, από τα αριστερά προς τα δεξιά. Εάν υπάρχουν ένθετες υπολογίζονται πρώτα οι εσωτερικές

**2 μοναδιαίοι τελεστές αύξησης μείωσης: ++, --**

Υπολογίζονται από δεξιά προς τα αριστερά.

**3 πολλαπλασιασμός, διαίρεση και υπόλοιπο: \*, /, ή %**

Υπολογίζονται από αριστερά προς τα δεξιά.

**4 πρόσθεση, αφαίρεση: + ή -**

Υπολογίζονται από τα αριστερά προς τα δεξιά.

**5 εκχώρησης: =, +=, -=, \*=, /=, %=**

Υπολογίζονται από δεξιά προς τα αριστερά.

⇒ Στην πράξη: να χρησιμοποιείται σχεδόν πάντα ()

# Μοναδιαίοι τελεστές αύξησης μείωσης

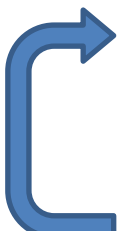
- Παραδείγματα

```
int a=8;  
int x = a--;  
cout << x <<"\n";  
cout << a <<"\n";
```

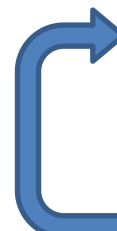
```
int a=8;  
int x = --a;  
cout << x <<"\n";  
cout << a <<"\n";
```

```
int a=5;  
cout << a <<"\n";  
cout << a++ <<"\n";  
cout << a <<"\n";
```

```
int a=5;  
cout << a <<"\n";  
cout << ++a <<"\n";  
cout << a <<"\n";
```



Πρώτα εκτυπώνει και μετά αυξάνει κατά 1



Πρώτα αυξάνει κατά 1 και μετά εκτυπώνει

Ενότητα 4

# **ΧΕΙΡΙΣΜΟΣ ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ, ΣΧΟΛΙΑ ΚΑΙ ΒΙΒΛΙΟΘΗΚΕΣ**

# Χειρισμός εισόδου/εξόδου

- Χειρισμός εισόδου

```
cin >> ...
```

- Χειρισμός εξόδου

```
cout << ...
```

- Βρίσκονται στη βιβλιοθήκη `iostream` και χρησιμοποιούν τον χώρο ονομάτων `std`

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

- Αν δεν είχαμε βάλει την εντολή `using namespace std` θα έπρεπε κάθε φορά που καλούμε τις `cout`, `cin` να γράφουμε:

```
std::cout << "Give a number" << "\n";  
std::cin >> number;
```

# Έξοδος με cout

- Εμφανίζει συνδυασμό μεταβλητών και αλφαριθμητικών " " (strings)

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

ή με μια εντολή:

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

- Μεταβλητές και αλφαριθμητικά:

```
cout << numberOfGames << " παιχνίδια παίχτηκαν.\n" ;
```

- Εκφράσεις: 

```
cout << "Συνολικό κόστος:" << (total+tax) << "\n" ;
```

- Μεταβλητές με κενά: 

```
cout << number1 << " " << number2;
```



# Αλλαγή γραμμής

- Χαρακτήρας αλλαγής γραμμής: "\n" (θεωρείται ως ένας χαρακτήρας)

```
cout << "Welcome\n\nto\n\nC++\n\nprogramming!!\n" ;
```

```
Welcome  
to  
C++  
programming!!
```

Για κάθε \n δημιουργεί  
μια αλλαγή γραμμής

- Άλλος τρόπος: endl

```
cout << "\n" ;
```

με εισαγωγικά

```
cout << endl ;
```

χωρίς εισαγωγικά

```
cout << "Έχετε κερδίσει ένα από \n"  
      << "τα παρακάτω βραβεία:\n";
```

```
cout << "Έχετε κερδίσει ένα από" << endl  
      << "τα παρακάτω βραβεία:" << endl;
```

# Χαρακτήρες διαφυγής

- Ειδικοί χαρακτήρες που ξεκινάνε με το σύμβολο \
  - Δεν έχουν την ίδια σημασία με τον χαρακτήρα χωρίς την κάθετο
- Αν θέλουμε να εκτυπώσουμε τα εισαγωγικά:

```
cout << "Your name is "John" " ;
```

ΛΑΘΟΣ

- Υπάρχει πρόβλημα διότι δεν γνωρίζει την αρχή και το τέλος του το αλφαριθμητικού " "

Ακολουθία	Σημασία
\n	Αλλαγή γραμμής
\t	Οριζόντιος στηλοθέτης
\a	Προειδοποίηση (ηχητικό καμπανάκι)
\\	Ανάποδη κάθετος \
\'	Μονό εισαγωγικό '
\"	Διπλά εισαγωγικά "

```
cout << "Very\tSimple\nExample" ;
```

```
Very    Simple  
Example
```

# Μορφοποίηση δεκαδικών αριθμών

```
cout << "Η τιμή είναι " << price << "€" << endl;
```

```
Η τιμή είναι 78.500000€
```

```
Η τιμή είναι  
78.50000e01€
```

```
Η τιμή είναι 78.5€
```

- Αν θέλουμε συγκεκριμένη μορφοποίηση στον αριθμό που εκτυπώνουμε π.χ. μόνο 2 δεκαδικά ψηφία μετά την υποδιαστολή:

1

```
cout.setf(ios::fixed);  
cout.setf(ios::showpoint);
```

Μόνο μια φορά γράφουμε τις εντολές αυτές

2

```
cout.precision(2);
```

Πριν την εκτύπωση του αριθμού

3

```
cout << "Η τιμή είναι " << price << "€" << endl;
```

```
Η τιμή είναι 78.50€
```

# Είσοδος με cin

- Διαβάζει από το πληκτρολόγιο κάποιες τιμές και αναθέτει σε μεταβλητές τις τιμές

## Σύνταξη

```
cin >> μεταβλητη1 >> μεταβλητη2 >> ...;
```

## Παραδείγματα

```
cin >> number >> size ;
```

```
cin >> number;
```

```
cout << "Δώσε έναν αριθμό: \n";  
cin >> number;
```

```
cout << "Δώσε τον αριθμό των δράκων\n"  
      << " και τον αριθμό των νάνων.";  
cin >> dragons >> trolls;
```

- Πολλαπλές τιμές όταν εισάγονται πρέπει να χωρίζονται με ένα ή περισσότερα κενά
  - Μόνο όταν πατηθεί το πλήκτρο Return γίνεται ανάθεση των τιμών

# Σχόλια

- Τα σχόλια στο πρόγραμμα δεν τα εξετάζει ο μεταφραστής και τα αγνοεί
  - Χρήση: για να γίνει ευανάγνωστο το πρόγραμμα

**1** // : οτιδήποτε ακολουθεί τα // είναι σχόλιο πάντα σε μια γραμμή

**2** /\* \*/ : οτιδήποτε περικλείεται από έναρξη /\* και τέλος \*/ είναι σχόλιο μπορεί να καλύψουν πολλές γραμμές

/\* Αυτό είναι ένα σχόλιο που καλύπτει τρεις γραμμές. Δεν υπάρχει κάποιο σύμβολο σχολίου ανάμεσα \*/

// Αυτό είναι ένα σχόλιο που // καλύπτει κάθε γραμμή // ξεχωριστά

# Βιβλιοθήκες

- Οδηγίες #include
- Γράφονται στην αρχή του προγράμματος

## Σύνταξη

```
#include <ονομαβιβλιοθηκης>
```

## Παραδείγματα

```
#include <iostream>
```

- Η βιβλιοθήκη `iostream` χρησιμοποιεί τις συναρτήσεις εισόδου/εξόδου `cin/cout`

- Ορισμένες άλλες
  - `cmath`
  - `cstdlib`

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double x;
    x = sqrt(9.0);
    cout << "Square of 9 is " << x;
}
```

Ενότητες 3 & 4

# ΠΑΡΑΔΕΙΓΜΑΤΑ

# Άθροισμα 2 ακεραίων

- Γράψτε ένα πρόγραμμα που διαβάζει δυο ακεραίους, υπολογίζει το άθροισμά τους και εκτυπώνει το άθροισμα



# Άθροισμα 2 ακεραίων

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

int main()
{
    int x,y;

    cout << "Give x: \n";
    cin >> x;
    cout << "Give y: \n";
    cin >> y;

    int sum = x + y;

    cout << "The sum of "<< x << " and " << y
          << " is "<< sum << endl;
    return 0;
}
```

# Ανταλλαγή 2 μεταβλητών

- Διαβάστε δύο ακεραίους και αποθηκεύστε τους σε δύο μεταβλητές τύπου int: x και y.

- Εκτυπώστε τις δύο μεταβλητές με την μορφή:

\ x = 78      y = 4 \

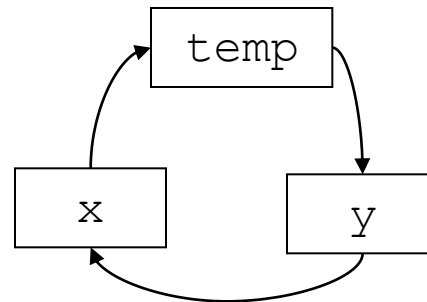
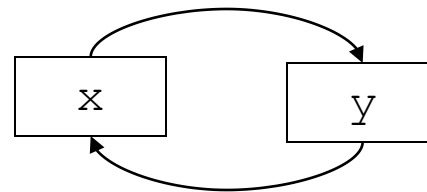
- Αλλάξτε τις τιμές των x και y. Δηλαδή η τιμή του x να έχει την τιμή του y, και η τιμή του y να έχει την τιμή του x.
- Ξαναεκτυπώστε τις δύο μεταβλητές με την μορφή:

\ x = 4      y = 78 \

# Παράδειγμα στις Μεταβλητές

- Αλλαγή (swap) των τιμών που αποθηκεύονται στις μεταβλητές x και y.
- Σκεφτείτε λίγο πώς θα μπορούσε να γίνει...
- Πρέπει να χρησιμοποιήσουμε μια νέα βοηθητική μεταβλητή.

```
int temp;  
temp = x;  
x = y;  
y = temp;
```



```

#include <iostream>
using namespace std;
int main()
{
    int x,y;

    cout << "Give x & y: \n";
    cin >> x >> y;

    cout << "\\ x = "<< x
         << "\t" << " y = " << y << " \\ \n";

    int temp;
    temp = x;
    x = y;
    y = temp;

    cout << "\\ x = "<< x
         << "\t" << " y = " << y << " \\ \n";
    return 0;
}

```

## Ανταλλαγή 2 μεταβλητών

Εκτυπώστε τις δύο μεταβλητές με την μορφή:  
 \ x = 78    y = 4 \

# Μετατροπή σε ώρες, λεπτά, δευτερόλεπτα

- Γράψτε ένα πρόγραμμα που δέχεται ως είσοδο έναν ακέραιο αριθμό που αναπαριστά χρονικό διάστημα σε δευτερόλεπτα. Το πρόγραμμα θα πρέπει να εμφανίζει στην έξοδο τον αριθμό των ωρών, των λεπτών, και των δευτερολέπτων που αντιστοιχούν στο χρονικό διάστημα δευτερολέπτων.
- Π.χ.,  
χρονικό διάστημα σε δευτερόλεπτα: 50391 (Είσοδος)

(Έξοδος): 13 ώρες

59 λεπτά

51 δευτερόλεπτα

# Μετατροπή σε ώρες, λεπτά, δευτερόλεπτα

```
int main()
{
    int time;
    int hours, mins, sec;

    cout << "Give time in sec: \n";
    cin >> time;

    hours = time / 3600;
    int minleft = time % 3600;
    mins = minleft / 60 ;
    sec = minleft % 60 ;

    cout << "hours: " << hours << endl
         << "min: " << mins << endl
         << "sec: " << sec << endl;
    return 0;
}
```

# **ΠΡΟΕΤΟΙΜΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ & DEV C++**

# Βήματα

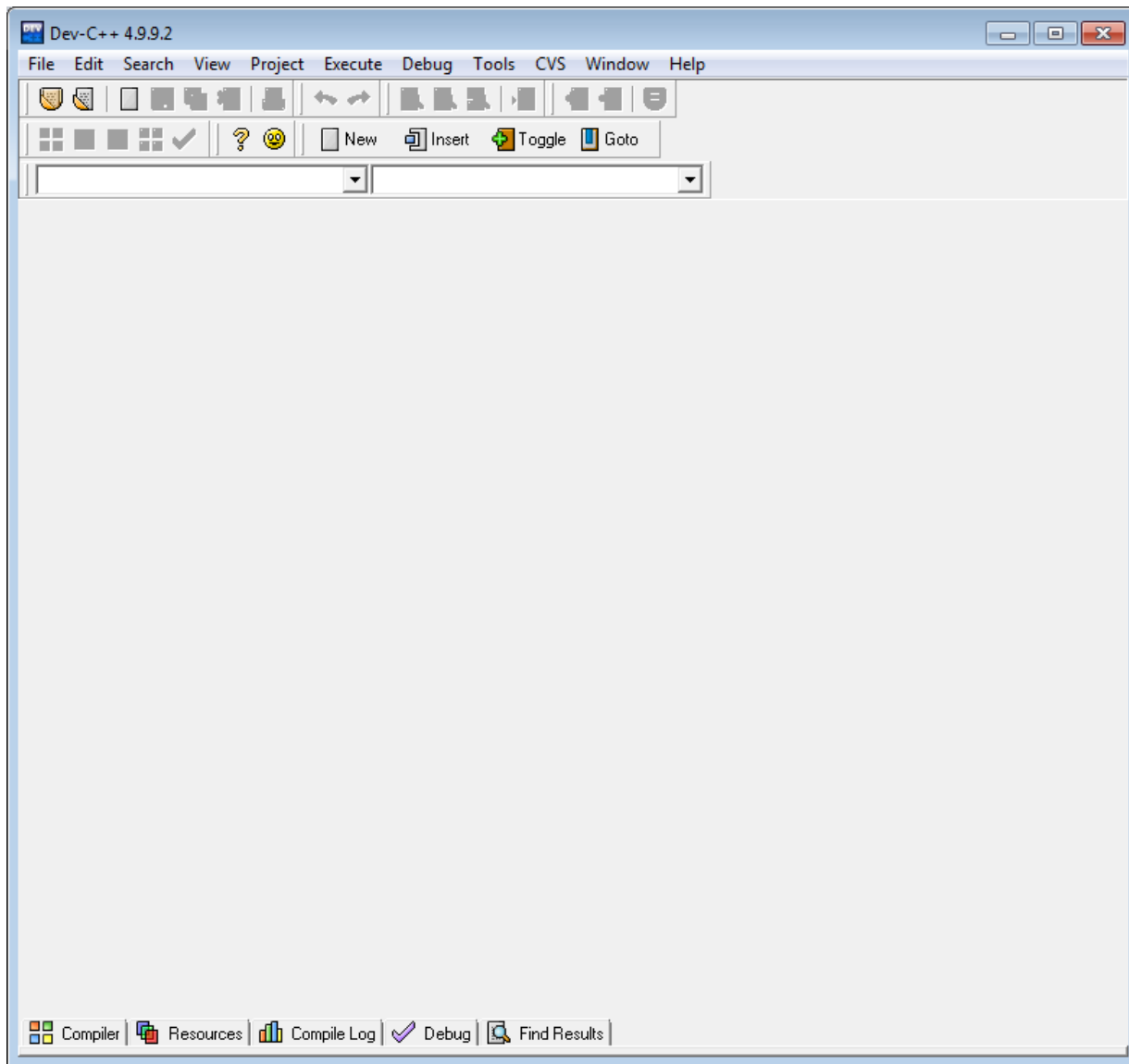
- Χρησιμοποιούμε τον μεταφραστή DevC++:

<http://www.bloodshed.net/dev/devcpp.html>

- 1 Αποθηκεύουμε το αρχείο με κατάληξη .cpp
  - 2 Γράφουμε τον κώδικα
  - 3 Μεταφράζουμε τον κώδικα (δηλαδή παράγουμε εκτελέσιμο αρχείο .exe) *προσοχή στα συντακτικά λάθη!*
  - 4 Εκτελούμε το πρόγραμμα
  - 5 Παρατηρούμε την εκτέλεση του προγράμματος και αν χρειαστεί εκτελούμε ξανά τα βήματα 2 3 4
- Όλα τα παραπάνω βήματα εκτελούνται μέχρι το επιθυμητό αποτέλεσμα

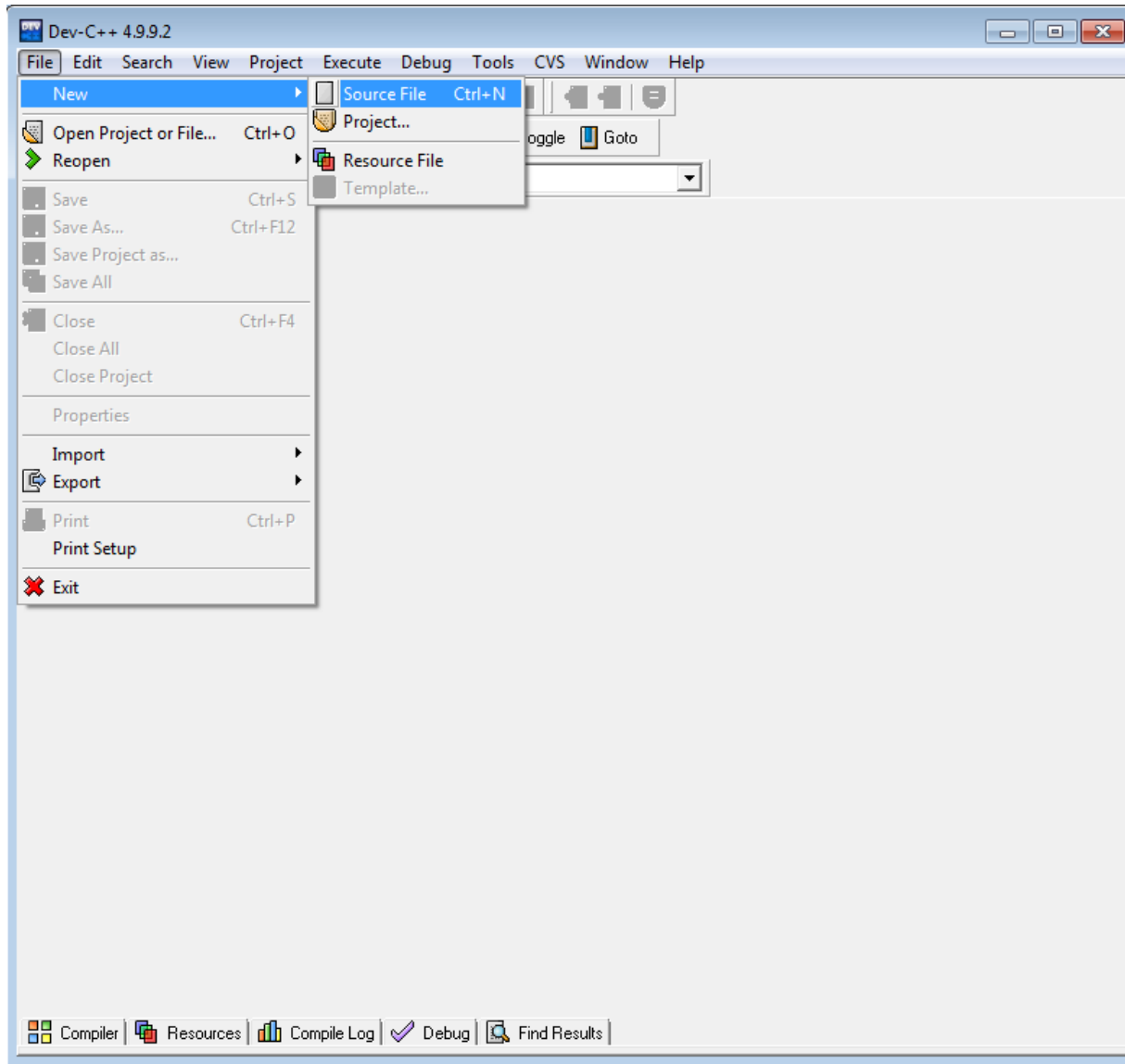


# Άνοιγμα πρώτου αρχείου



- Ανοίγουμε το devc++

# Άνοιγμα πρώτου αρχείου




- Δημιουργούμε ένα νέο αρχείο (source file)

# Άνοιγμα πρώτου αρχείου

- Όταν δημιουργούμε ένα αρχείο:
  - Αν υπάρχουν έτοιμες εντολές δεν τις πειράζουμε
  - Αν δεν υπάρχουν πρέπει να τις προσθέσουμε

Στο σημείο  
αυτό  
γράφουμε τον  
κώδικα που  
θέλουμε



```
#include <iostream>

using namespace std;

int main( )
{

    system("PAUSE");
    return 0;

}
```

# Παράδειγμα

- Διάβασμα ενός ακεραίου και εκτύπωση του αντίθετού του

$$x = 45 \Rightarrow x = -45$$

$$x = -8 \Rightarrow x = 8$$

# Παράδειγμα

- Διάβασμα ενός ακεραίου και εκτύπωση του αντίθετού του

$x = 45 \Rightarrow x = -45$

$x = -8 \Rightarrow x = 8$

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

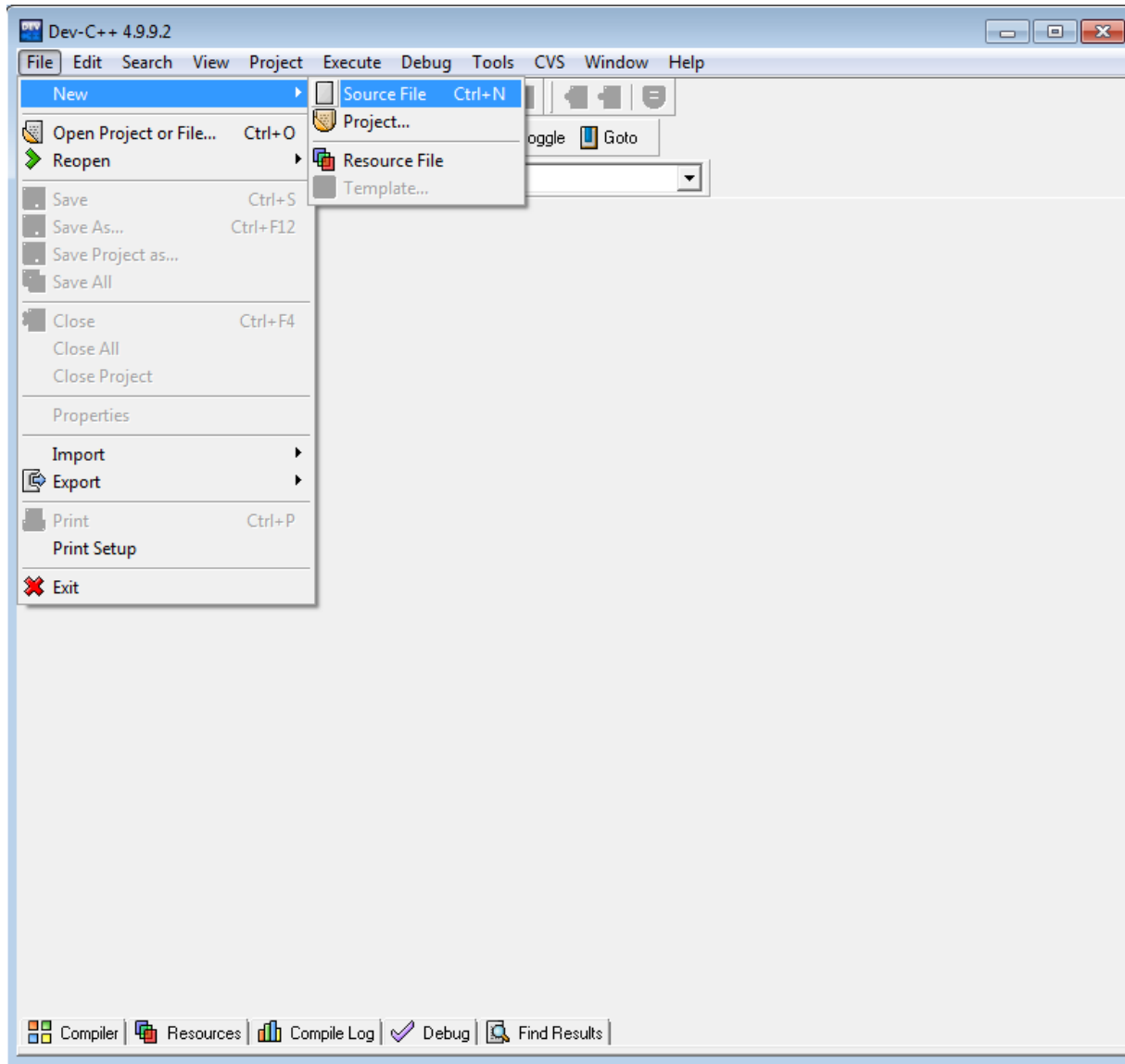
int main()
{
    int x;

    cin >> x;

    x = -1 * x;    // x=-x ?

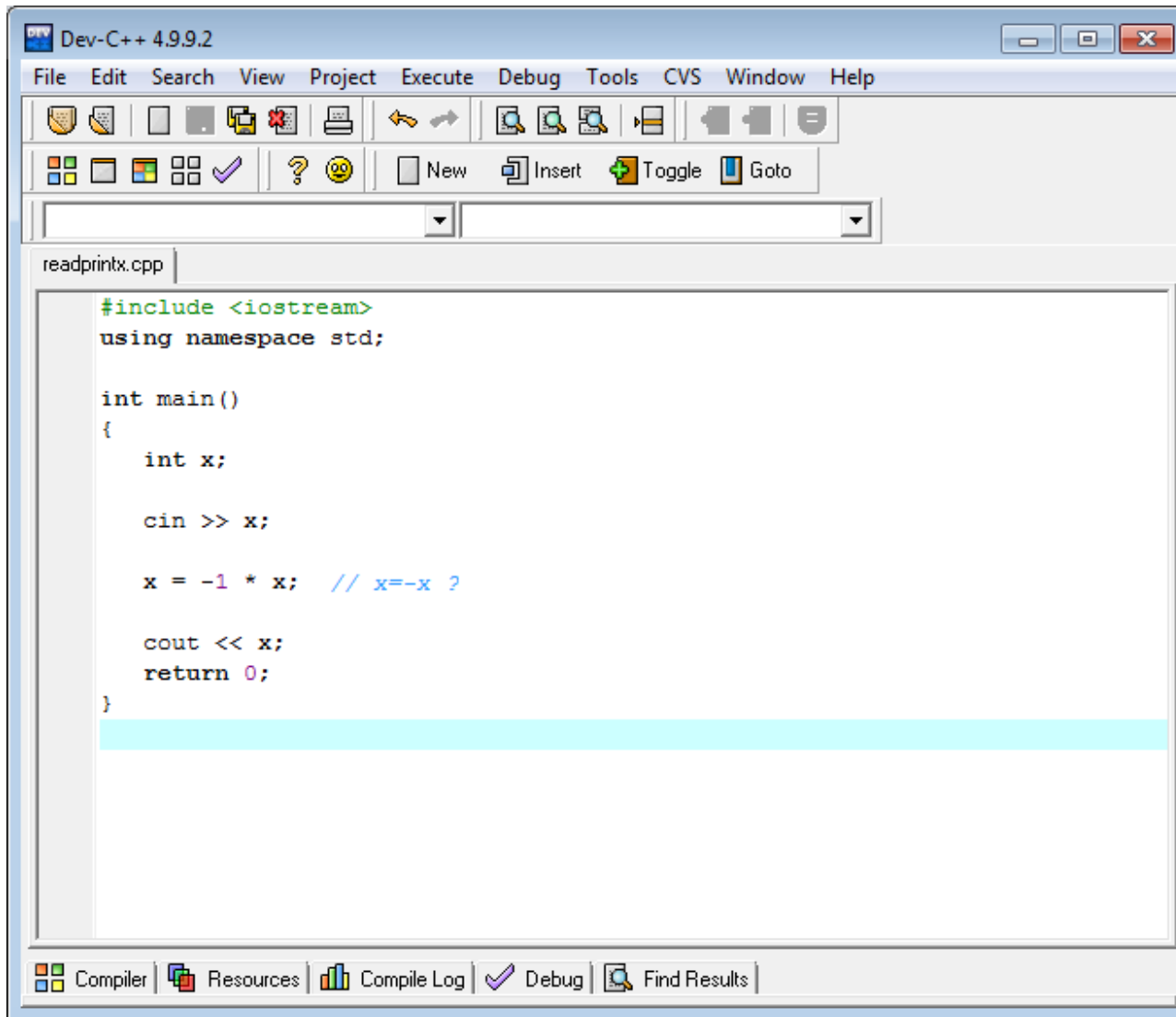
    cout << x;
    return 0;
}
```

# Άνοιγμα πρώτου αρχείου



- Δημιουργούμε ένα νέο αρχείο (source file)

# Αποθήκευση του αρχείου



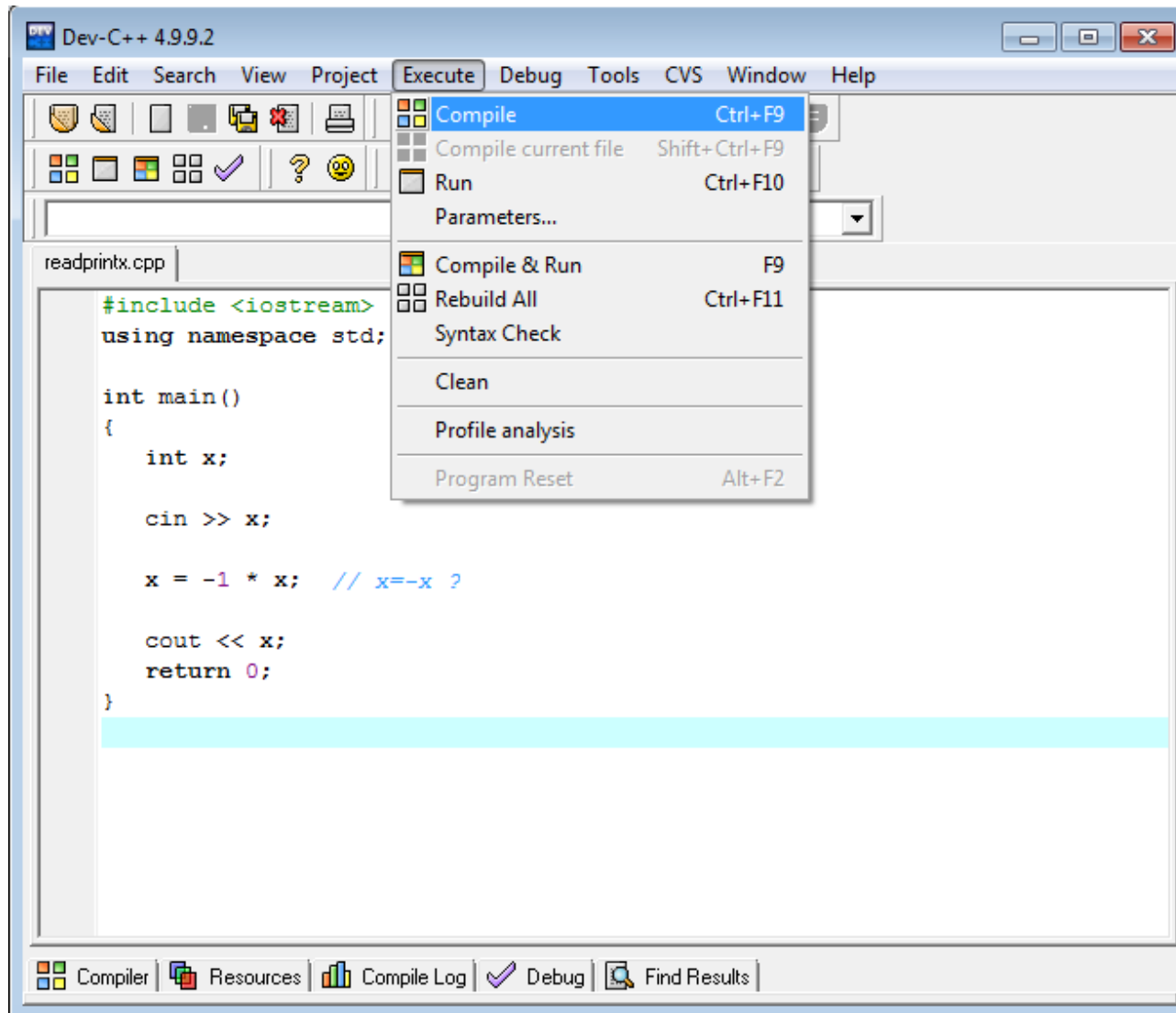
```
readprintx.cpp  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int x;  
  
    cin >> x;  
  
    x = -1 * x; // x=-x ?  
  
    cout << x;  
    return 0;  
}
```

- Αρχείο με κατάληξη .cpp  
π.χ. lab1.cpp,  
askisi2.cpp, ...

## Εργαστήρια:

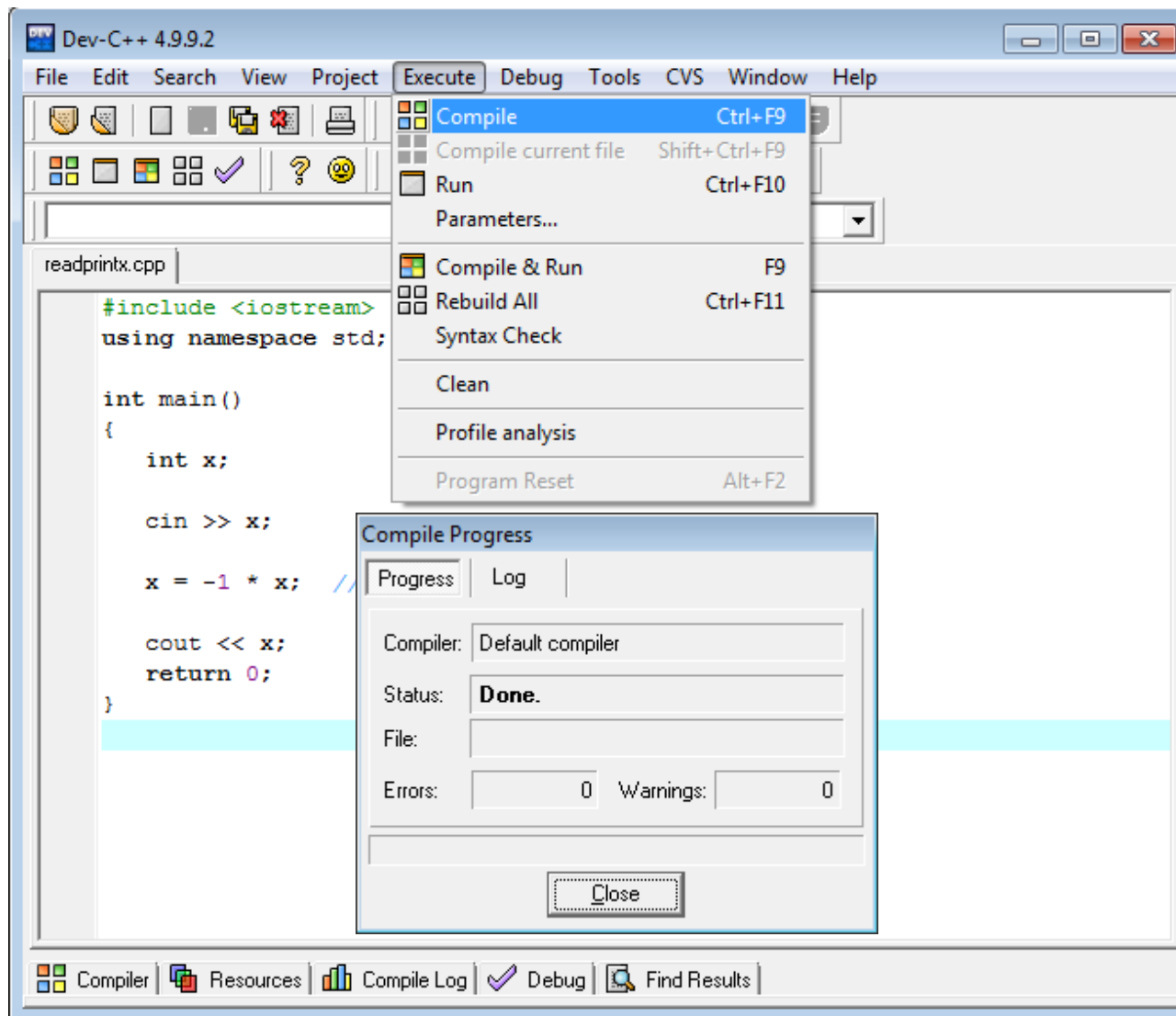
- Αποθήκευση πάντα μέσα στο φάκελο με Αριθμ. Μητρ. (π.χ. 10099) που έχετε δημιουργήσει στο "userspace"

# Μετάφραση (compile)





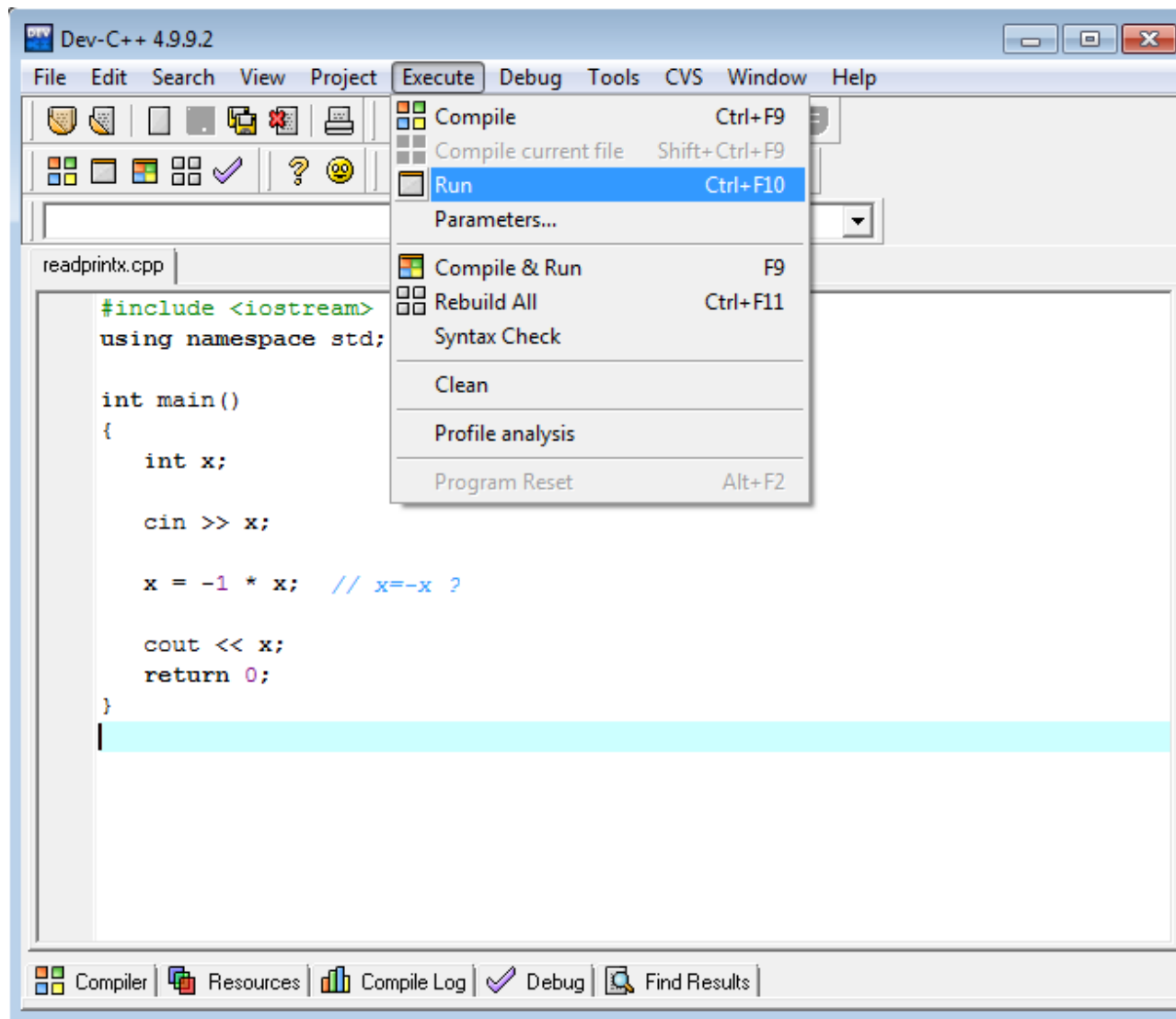
# Μετάφραση (compile)



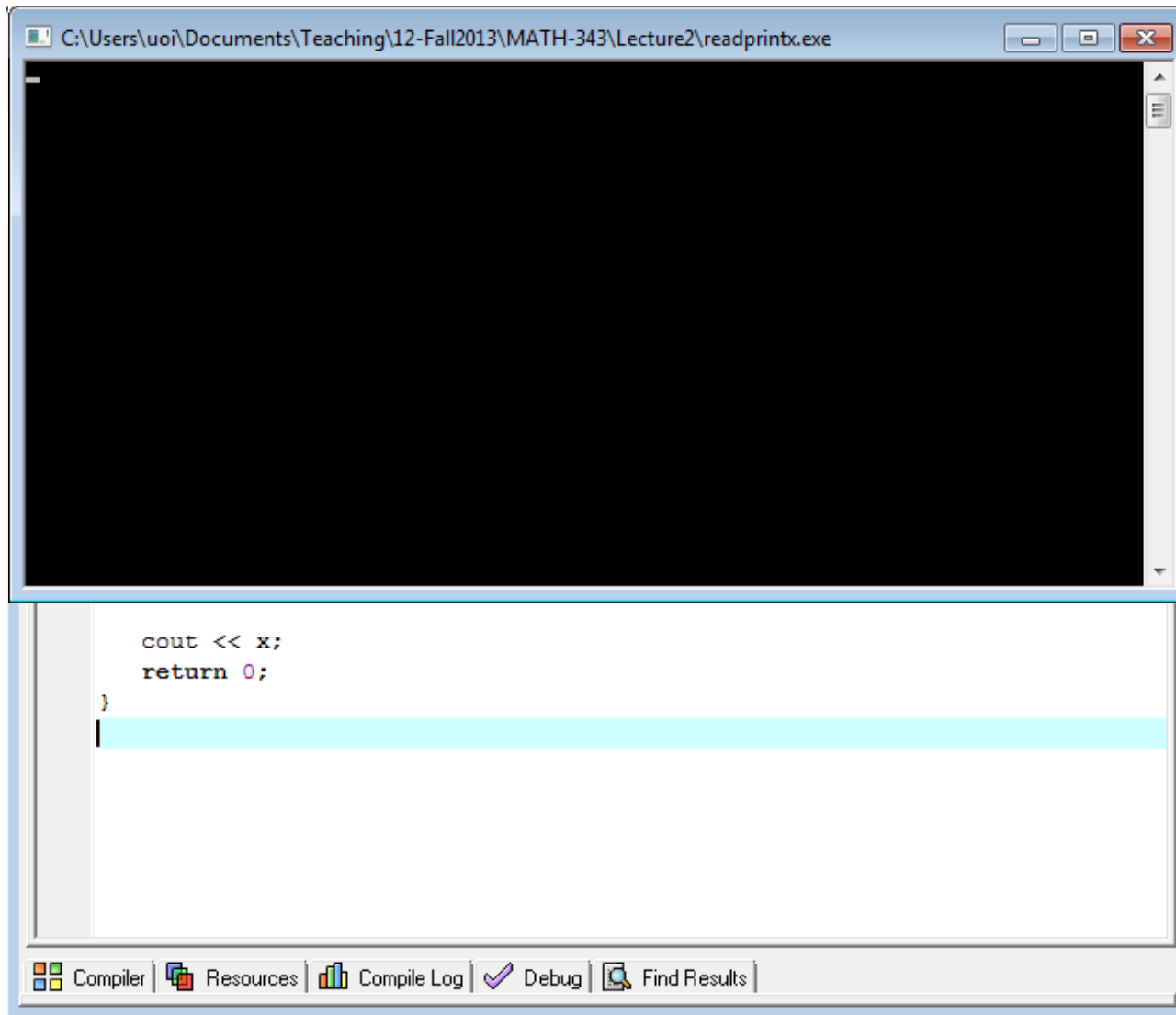
Μήνυμα αποτελέσματος:

- Εξετάζουμε τα πιθανά λάθη

# Εκτέλεση

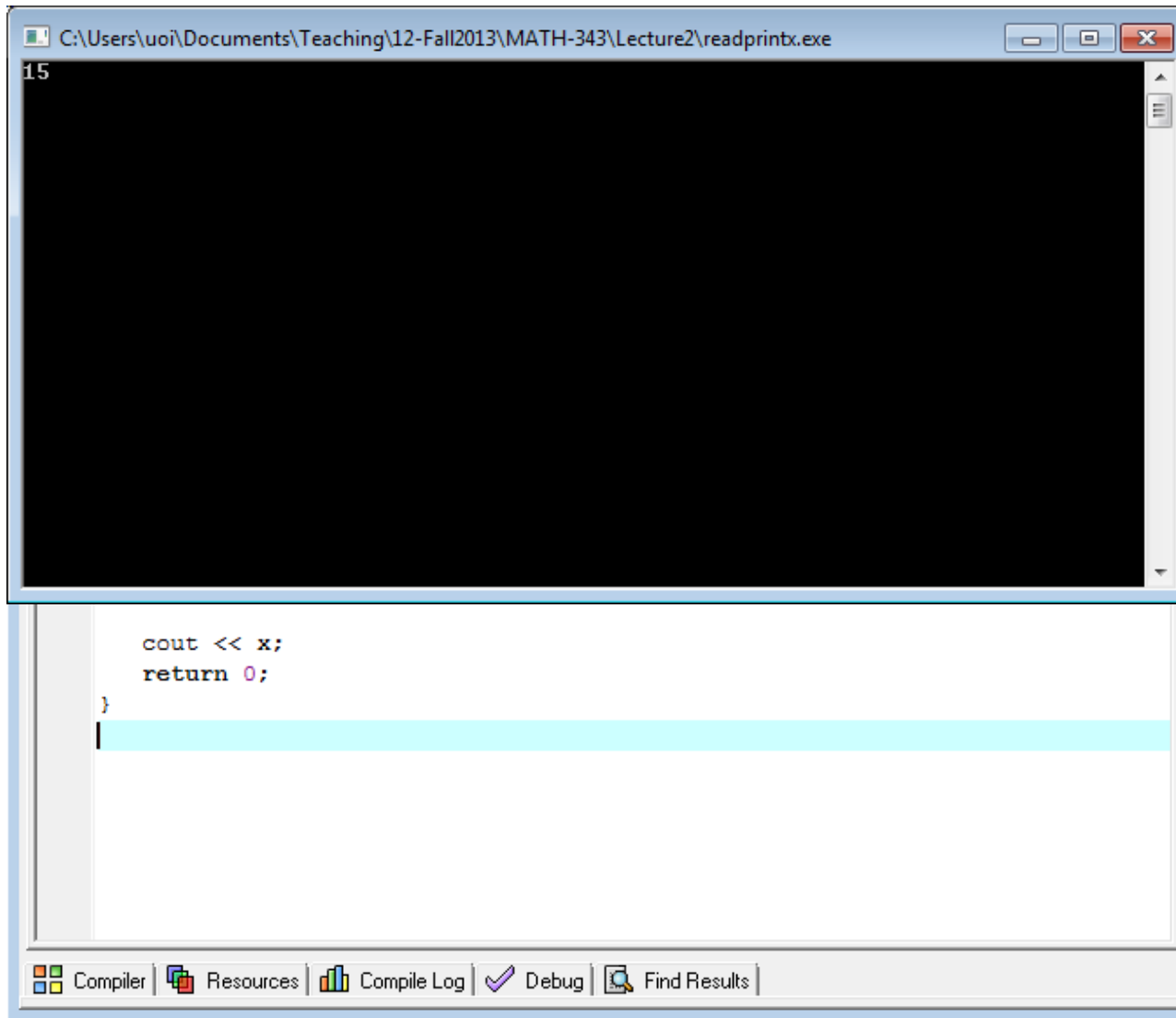


# Εκτέλεση



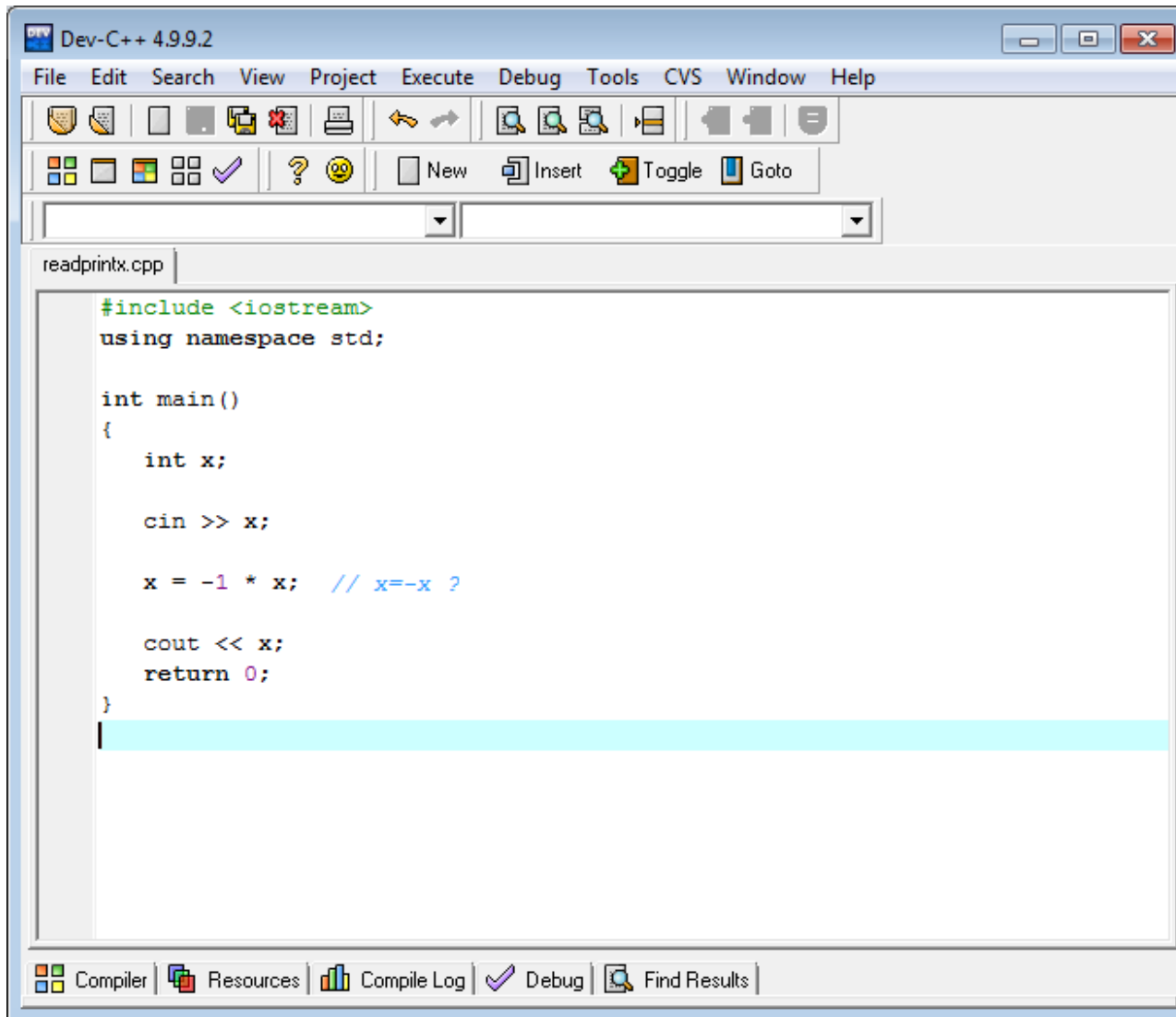
- Τι περιμένει;

# Εκτέλεση



- Τι περιμένει;

# Πρόβλημα στην Εκτέλεση



The screenshot shows the Dev-C++ 4.9.9.2 IDE. The main window displays a C++ program named 'readprintx.cpp'. The code is as follows:

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

int main()
{
    int x;

    cin >> x;

    x = -1 * x; // x=-x ?

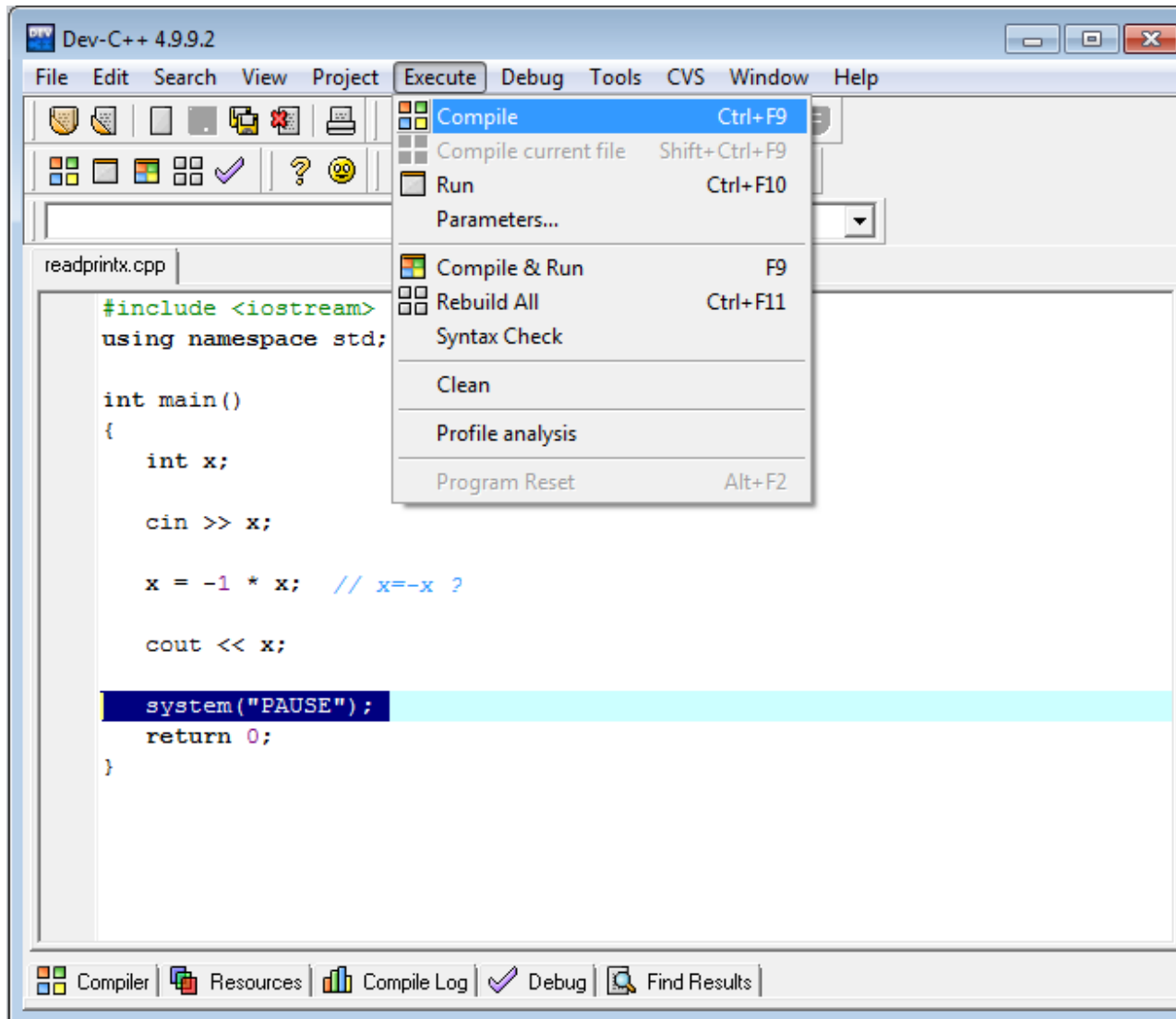
    cout << x;
    return 0;
}
```

The line `x = -1 * x; // x=-x ?` is highlighted in light blue. The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, Help), a toolbar with various icons, and a status bar at the bottom with tabs for Compiler, Resources, Compile Log, Debug, and Find Results.

- Εξαφανίστηκε το παράθυρο!
- Πού είναι το αποτέλεσμά μας;

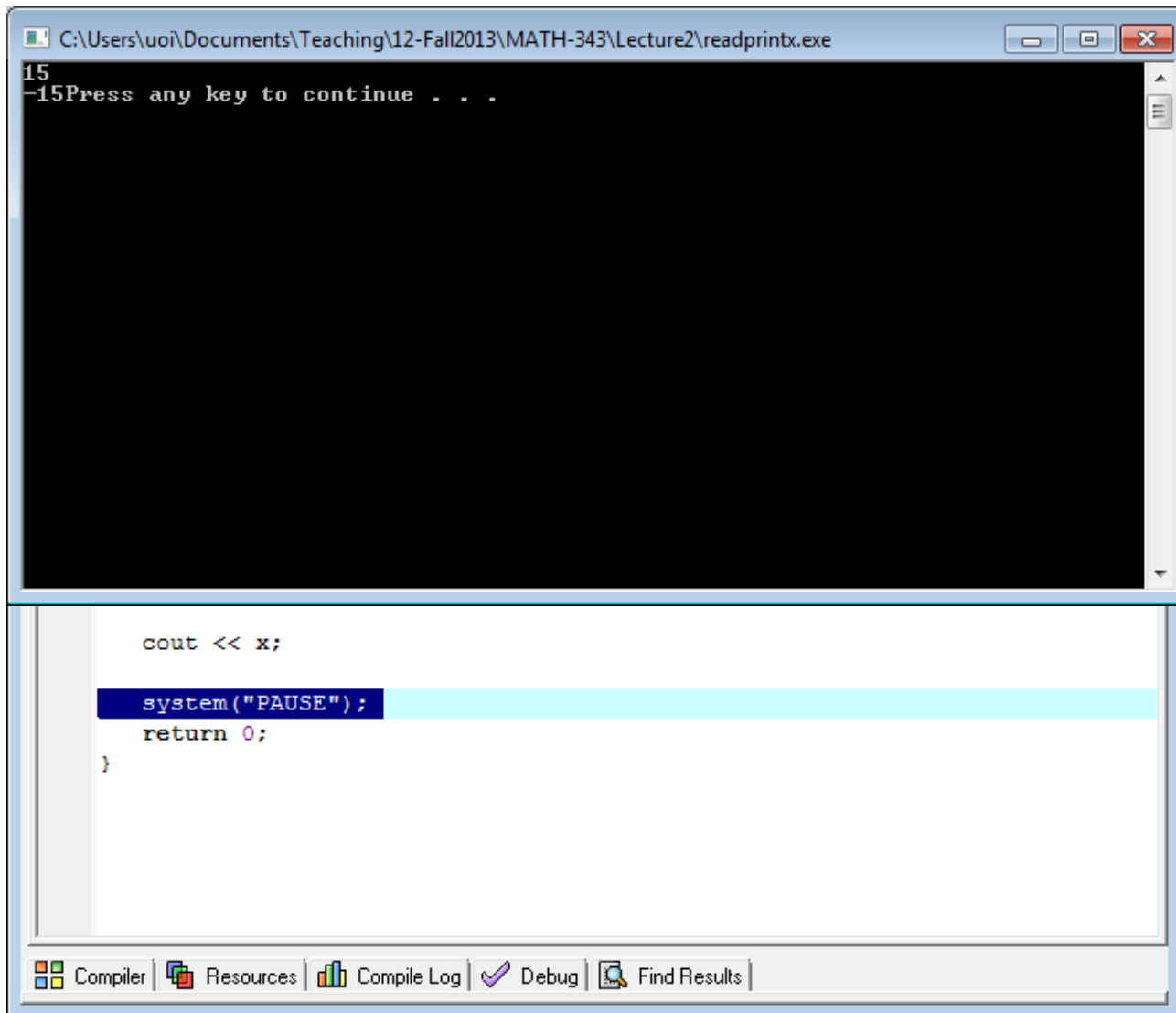


# Πρόβλημα στην Εκτέλεση



- Εξαφανίστηκε το παράθυρο!
  - Πού είναι το αποτέλεσμά μας;
  - Προσθέσαμε την εντολή...  
    τώρα;
- ξανά μετάφραση!!

# Ξανά εκτέλεση



```
C:\Users\uoι\Documents\Teaching\12-Fall2013\MATH-343\Lecture2\readprintx.exe
15
-15Press any key to continue . . .

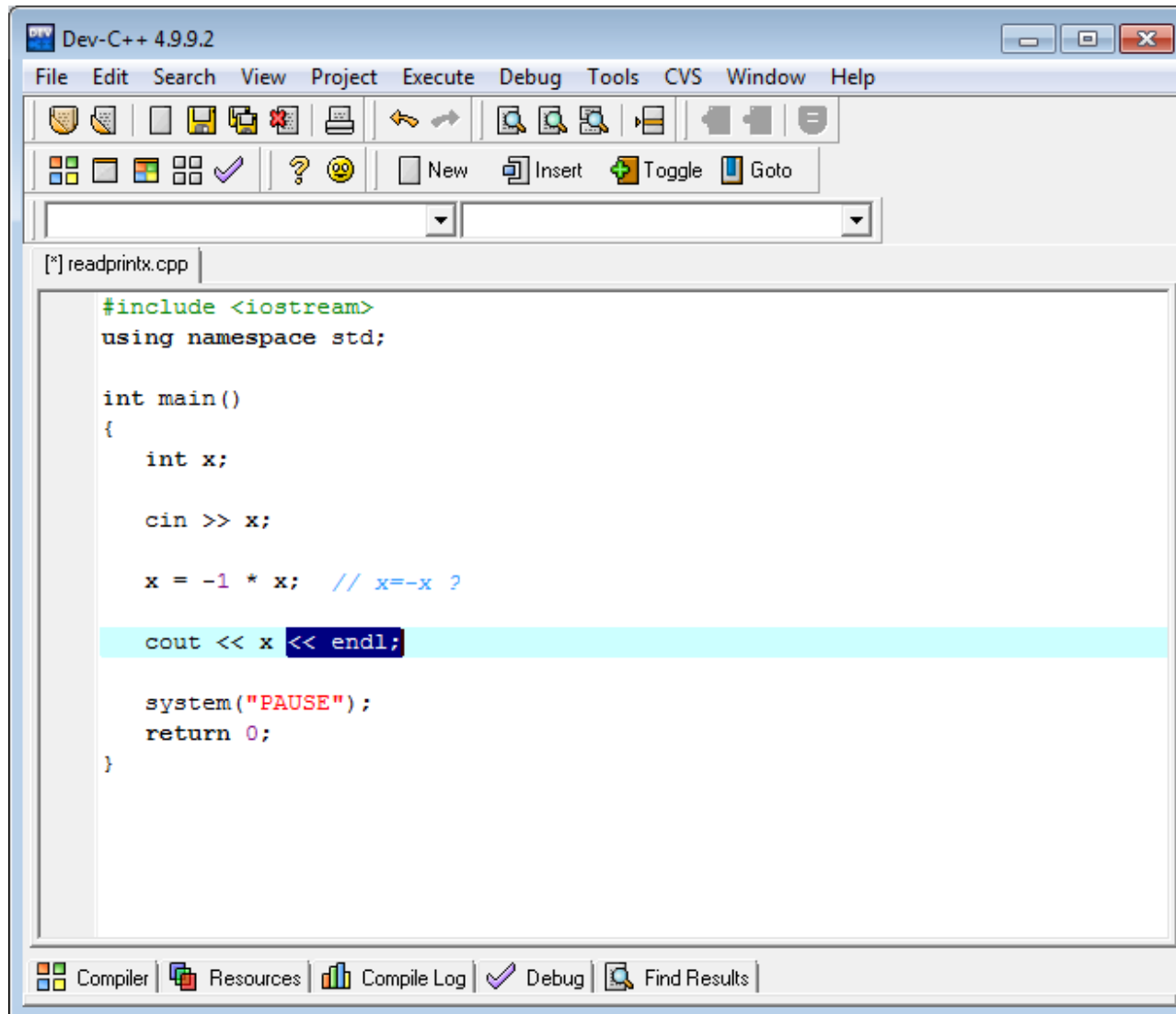
cout << x;
system("PAUSE");
return 0;
}
```

Compiler | Resources | Compile Log | Debug | Find Results

- Σωστό αποτέλεσμα
- Κάτι δεν μας αρέσει



# Ξανά εκτέλεση



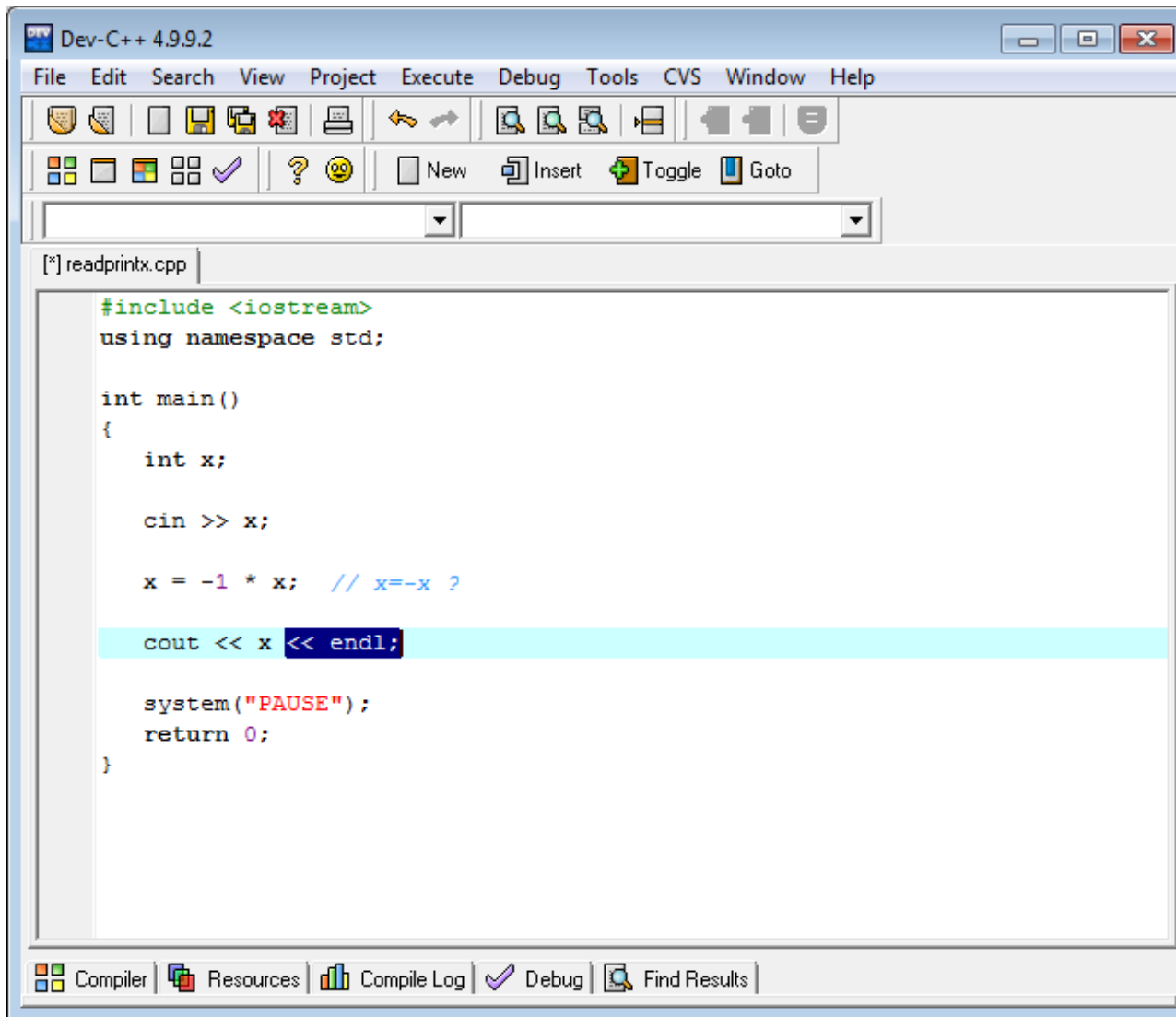
The screenshot shows the Dev-C++ 4.9.9.2 IDE interface. The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. The main editor window displays the following C++ code:

```
[*] readprintx.cpp |  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int x;  
  
    cin >> x;  
  
    x = -1 * x; // x=-x ?  
  
    cout << x << endl;  
  
    system("PAUSE");  
    return 0;  
}
```

The line `cout << x << endl;` is highlighted in light blue. The status bar at the bottom shows icons for Compiler, Resources, Compile Log, Debug, and Find Results.

- Αλλαγή γραμμής στο τέλος!

# Ξανά εκτέλεση



The screenshot shows the Dev-C++ 4.9.9.2 IDE. The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains icons for file operations, editing, and execution. The main editor window displays the following C++ code:

```
[*] readprintx.cpp  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int x;  
  
    cin >> x;  
  
    x = -1 * x; // x=-x ?  
  
    cout << x << endl;  
  
    system("PAUSE");  
    return 0;  
}
```

The line `cout << x << endl;` is highlighted in light blue. The status bar at the bottom shows icons for Compiler, Resources, Compile Log, Debug, and Find Results.

- Αλλαγή γραμμής στο τέλος!

- Ξανά:
  - 3 compile
  - 4 run

# Ξανά εκτέλεση

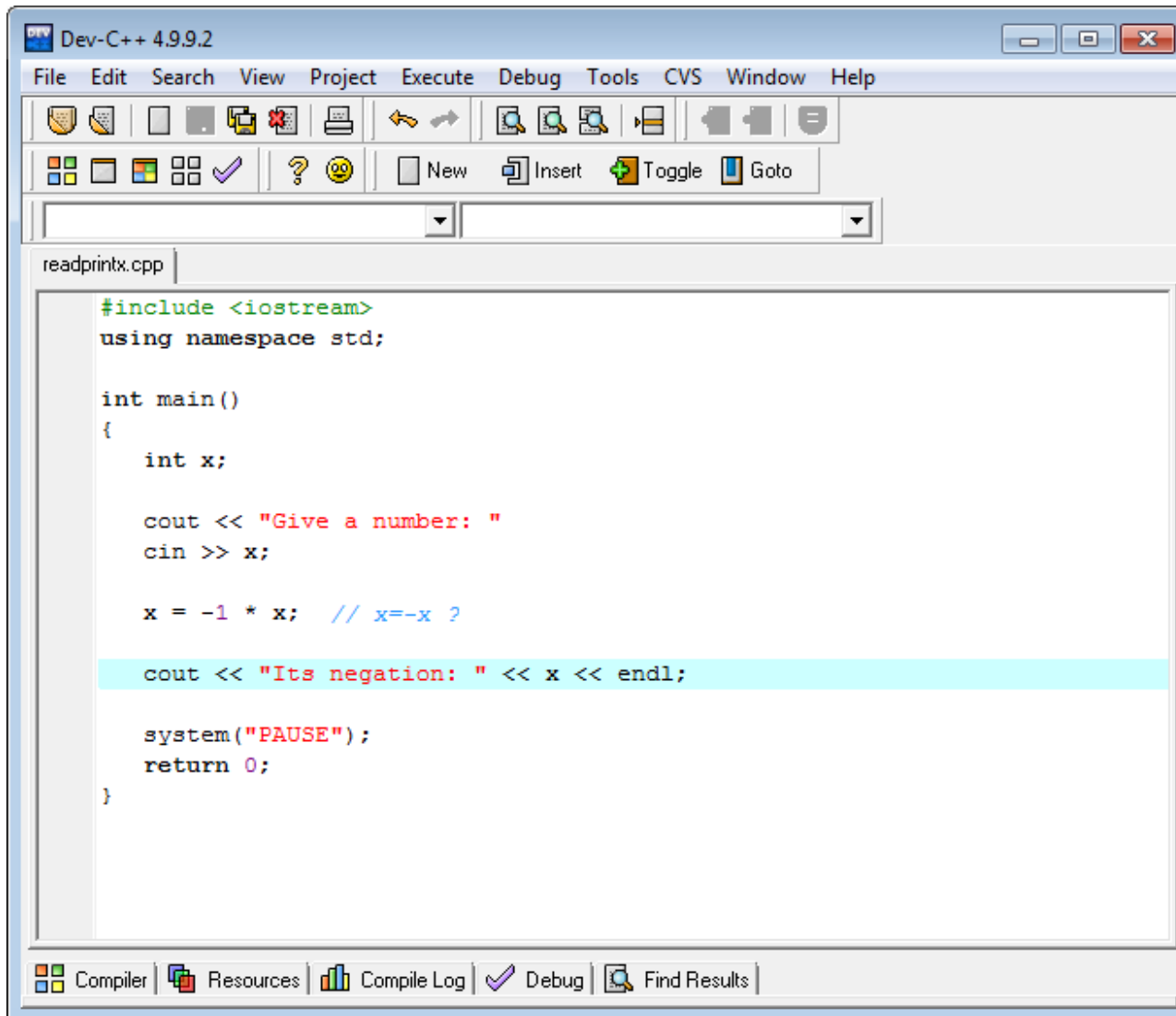
```
C:\Users\uoi\Documents\Teaching\12-Fall2013\MATH-343\Lecture2\readprintx.exe
15
-15
Press any key to continue . . .

cout << x << endl;

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

- Αλλαγή γραμμής στο τέλος!
- Ξανά:
  - 3 compile
  - 4 run
- Ας βγαινουν και κάποια μηνύματα...

# Διόρθωση



The screenshot shows the Dev-C++ 4.9.9.2 IDE. 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 window displays the following C++ code:

```
readprintx.cpp |
#include <iostream>
using namespace std;

int main()
{
    int x;

    cout << "Give a number: "
    cin >> x;

    x = -1 * x; // x=-x ?

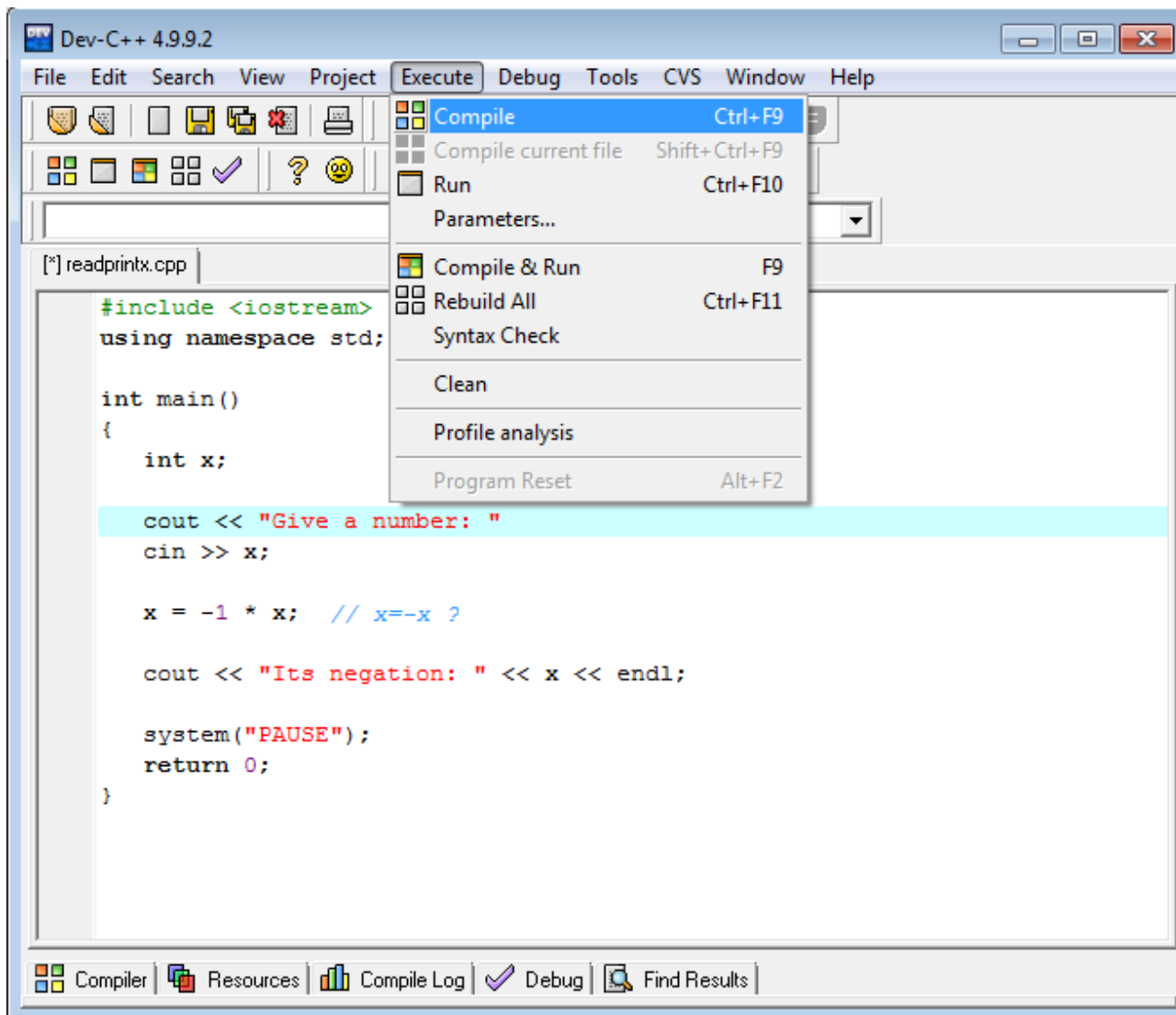
    cout << "Its negation: " << x << endl;

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

The line `cout << "Its negation: " << x << endl;` is highlighted in light blue. The status bar at the bottom shows Compiler, Resources, Compile Log, Debug, and Find Results.

- Πρόσθεση μηνυμάτων

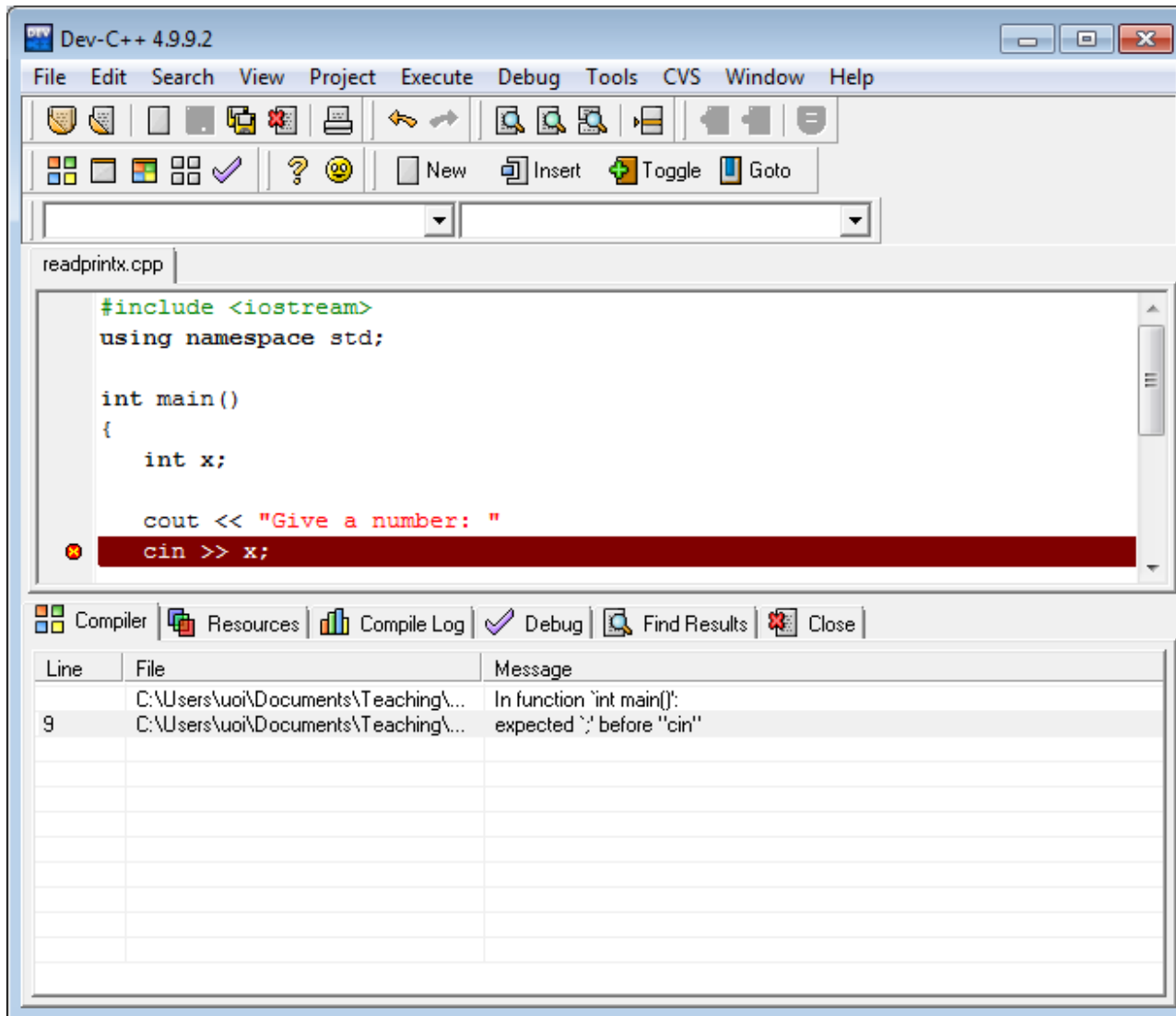
# Διόρθωση



- Πρόσθεση μηνυμάτων

3 compile

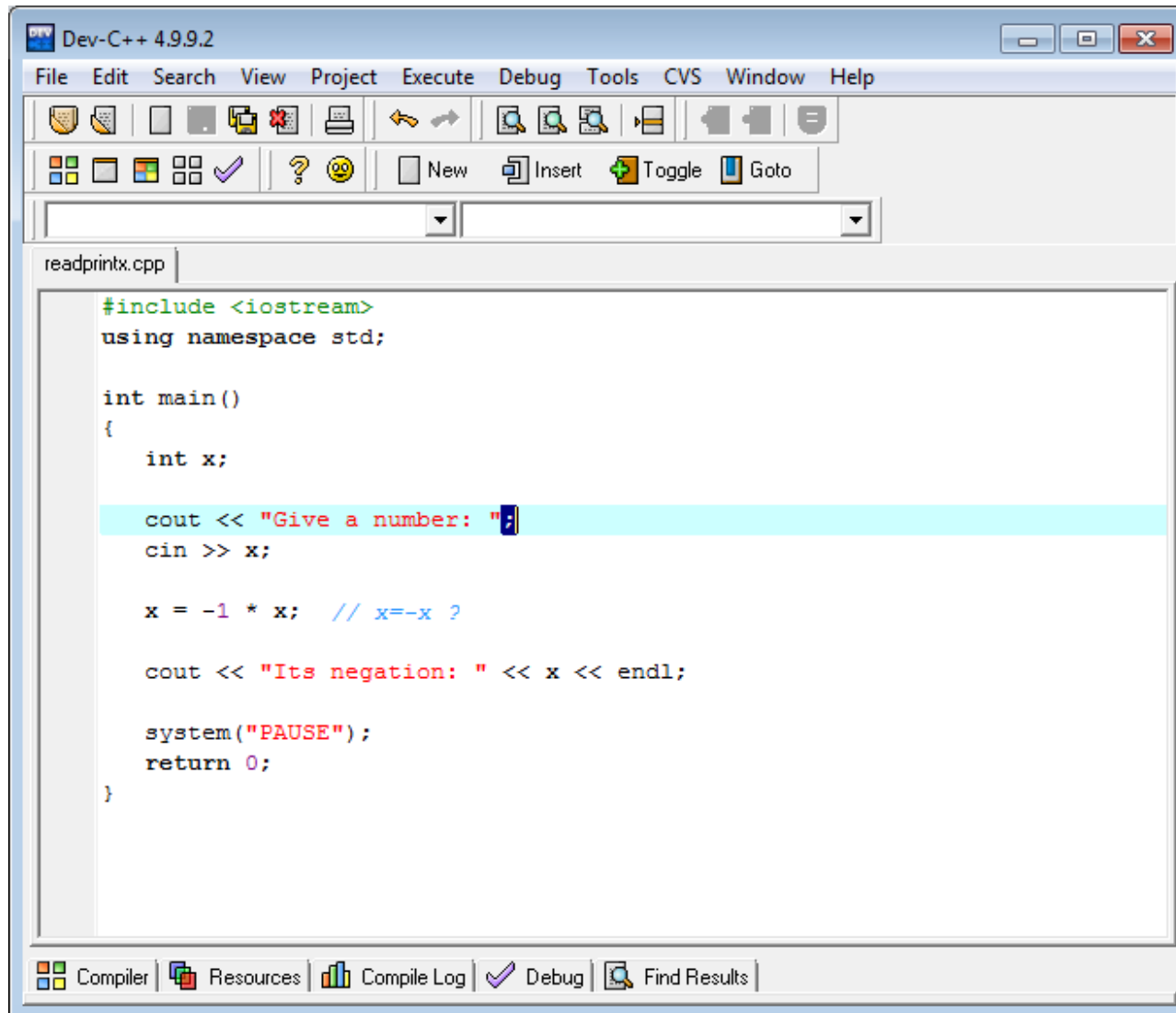
# Διόρθωση



- Λάθη κατά την μετάφραση!!!

Τα μηνύματα που βγαίνουν σας βοηθάνε να καταλάβετε τι είναι λάθος

# Διόρθωση



The image shows 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 window displays the source code for "readprintx.cpp". The code is as follows:

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

int main()
{
    int x;

    cout << "Give a number: ";
    cin >> x;

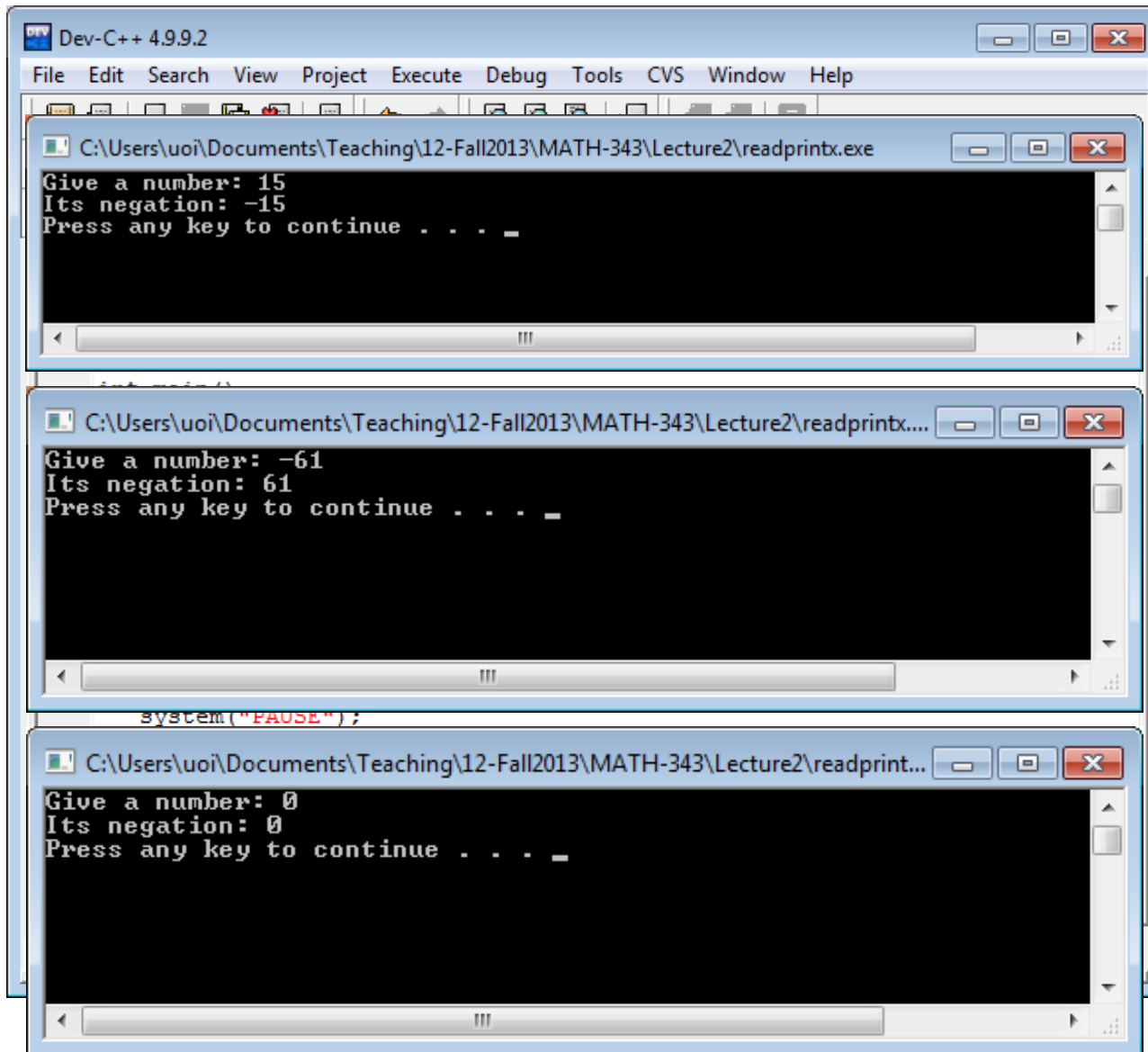
    x = -1 * x; // x=-x ?

    cout << "Its negation: " << x << endl;

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

The line `cout << "Give a number: ";` is highlighted in light blue. The status bar at the bottom shows "Compiler", "Resources", "Compile Log", "Debug", and "Find Results".

# Εκτέλεση



```
Dev-C++ 4.9.9.2
File Edit Search View Project Execute Debug Tools CVS Window Help
C:\Users\uoι\Documents\Teaching\12-Fall2013\MATH-343\Lecture2\readprintx.exe
Give a number: 15
Its negation: -15
Press any key to continue . . . . _

C:\Users\uoι\Documents\Teaching\12-Fall2013\MATH-343\Lecture2\readprintx...
Give a number: -61
Its negation: 61
Press any key to continue . . . . _

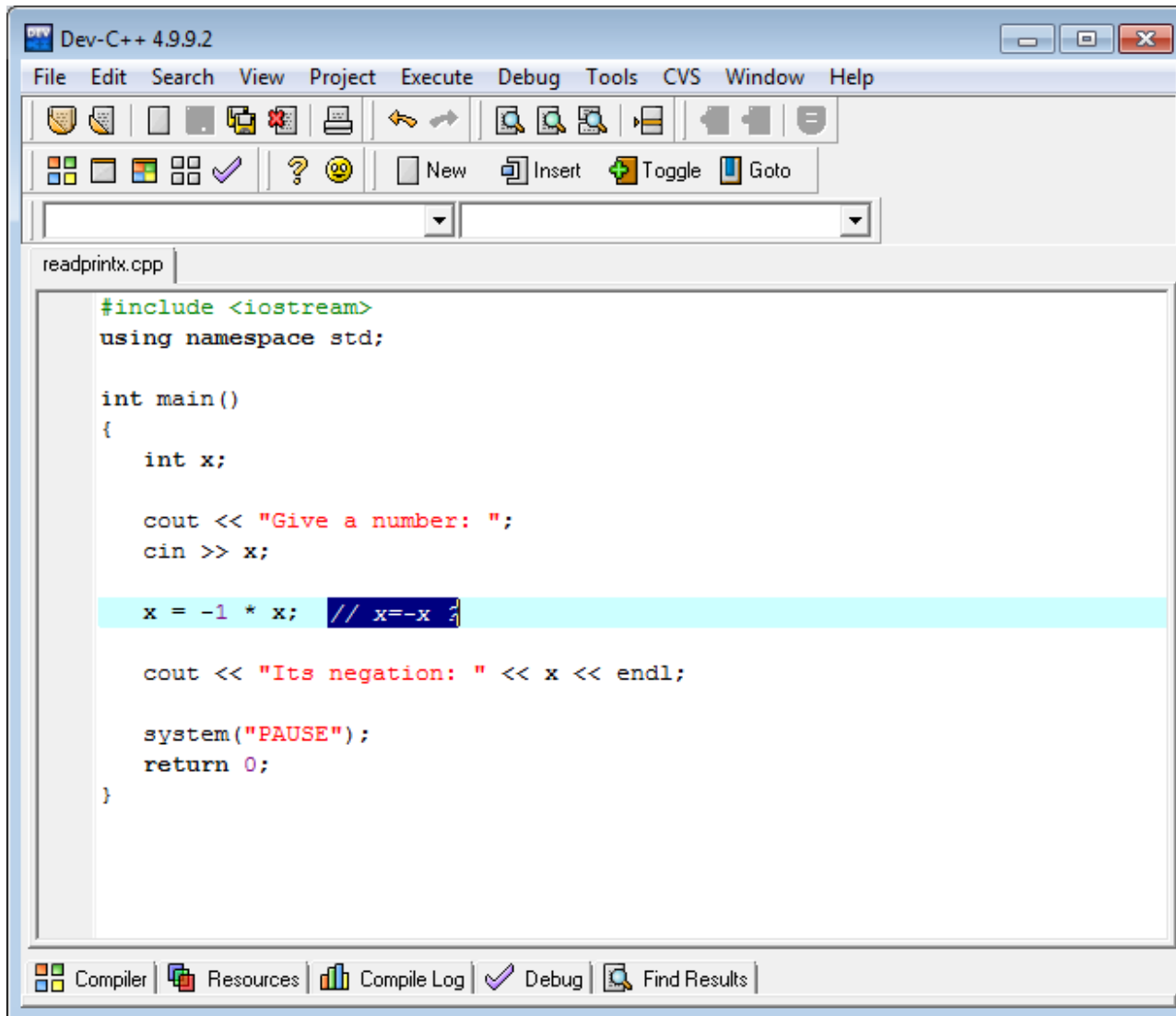
system("PAUSE");

C:\Users\uoι\Documents\Teaching\12-Fall2013\MATH-343\Lecture2\readprint...
Give a number: 0
Its negation: 0
Press any key to continue . . . . _
```

- Έλεγχο αρκετών εκτελέσεων



# Ερώτημα



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

int main()
{
    int x;

    cout << "Give a number: ";
    cin >> x;

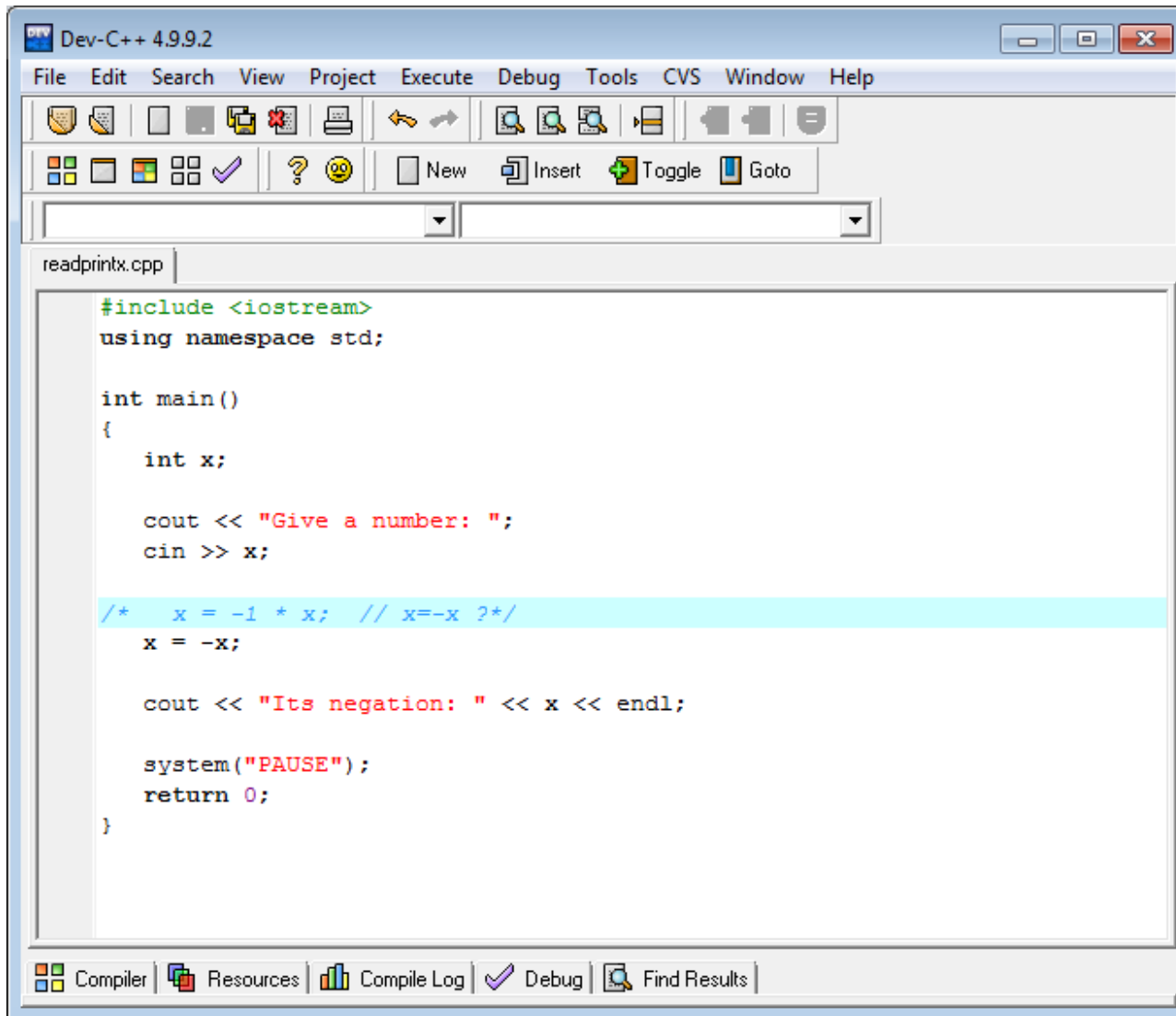
    x = -1 * x; // x=-x

    cout << "Its negation: " << x << endl;

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

- Μήπως δουλεύει η εντολή  
 $x = -x;$   
αντί για  
 $x = -1 * x;$

# Ερώτημα



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

int main()
{
    int x;

    cout << "Give a number: ";
    cin >> x;

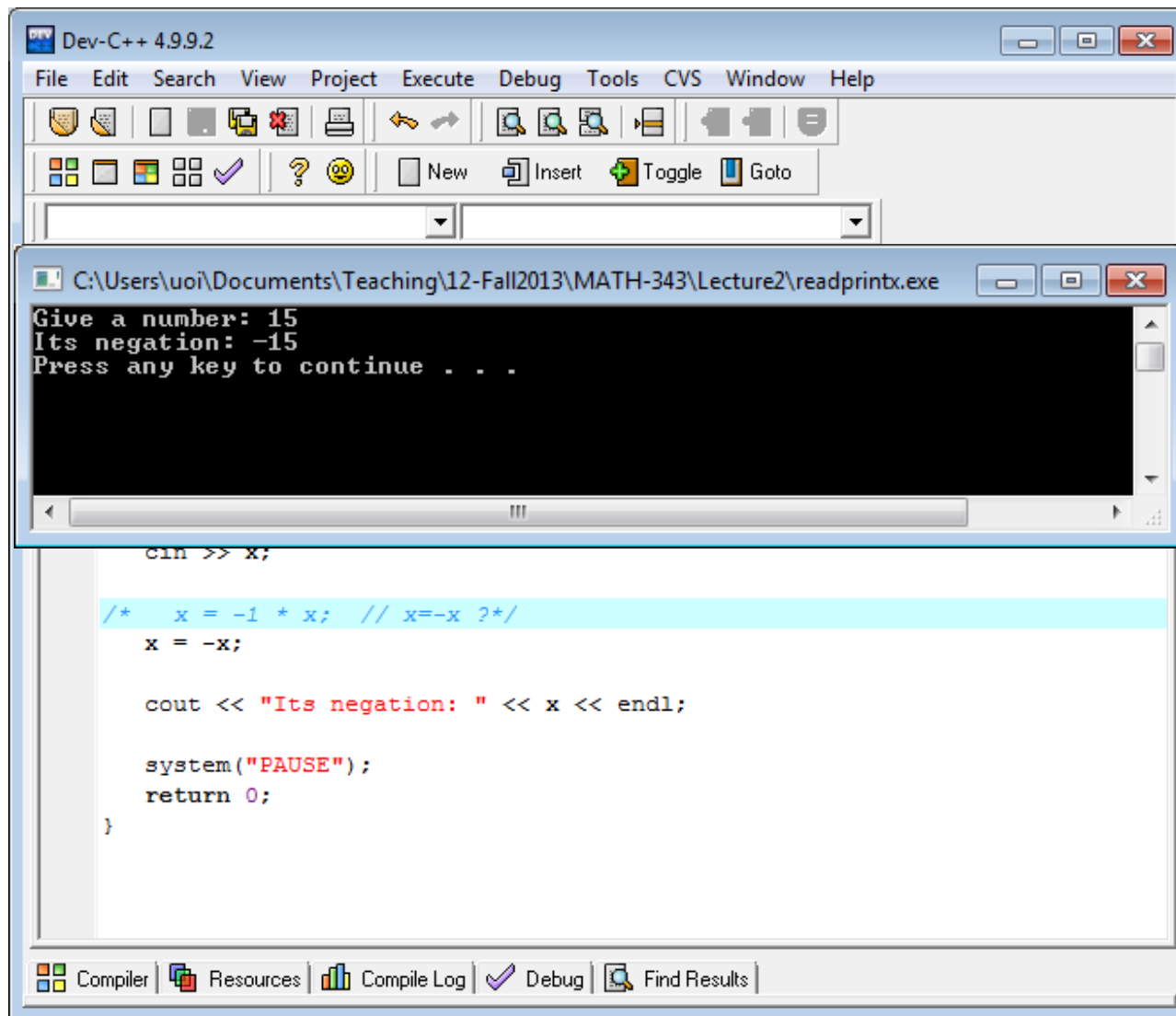
    /* x = -1 * x; // x=-x ?*/
    x = -x;

    cout << "Its negation: " << x << endl;

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

- Μήπως δουλεύει η εντολή  
 $x = -x;$   
αντί για  
 $x = -1 * x;$
- Δοκιμάστε το!!

# Ερώτημα



The screenshot shows the Dev-C++ 4.9.9.2 IDE. The top window displays the execution output of a program. The bottom window shows the source code of the program.

```
Give a number: 15
Its negation: -15
Press any key to continue . . .
```

```
cin >> x;

/* x = -1 * x; // x=-x ?*/
x = -x;

cout << "Its negation: " << x << endl;

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

- Μήπως δουλεύει η εντολή  
 $x = -x;$   
αντί για  
 $x = -1 * x;$
- Δοκιμάστε το!!

# 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).

# Εργαστήρια & Τεστ Quiz

- Στα Εργαστήρια θα πρέπει:
  - Να κατανοείτε το πρόβλημα και να προσπαθείτε να σχεδιάσετε την λύση **πριν έλθετε στο εργαστήριο**. → PreLab.pdf → PreLab-1.pdf
  - Να συμμετέχετε ενεργά και να είστε **προετοιμασμένοι** να απαντάτε σε ερωτήσεις σχετικές με την εκφώνηση → Lab.pdf → Lab-1.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).

# PreLab-1.pdf

- ΠΡΙΝ το 1<sup>ο</sup> Εργαστήριο θα πρέπει να ασχοληθείτε με τα (απλά) ζητήματα.
- Απορίες λύνονται κατά τη διάρκεια του Εργαστηρίου
- Στο 1<sup>ο</sup> Εργαστήριο θα πρέπει να αναπτύξετε κώδικα που θα σας δυσκολέψει αν δεν ασχοληθείτε με PreLab-1.pdf
- Δεν χρειάζεστε τη λύση τους
  - αρκεί να τα κατανοήσετε

## 343 Εισαγωγή στον Προγραμματισμό :

### ΠΡΟΕΤΟΙΜΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ & ΘΕΜΑΤΑ ΚΑΤΑΝΟΗΣΗΣ

#### 1<sup>ο</sup> Εργαστήριο

##### Χρήσιμο Υλικό:

- Διαφάνειες 2<sup>ης</sup> εβδομάδας & Εισαγωγικά για το εργαστήριο
- Βοηθητικό αρχείο: `readprintX.cpp` *διάβασμα ακεραίου και εκτύπωση του διπλάσιου ακεραίου*

##### Ζήτημα 1<sup>ο</sup>

Δημιουργήστε ένα πρόγραμμα που θα διαβάζει έναν ακέραιο  $x$ , θα υπολογίζει το διπλάσιό του ( $2x$ ), θα υπολογίζει τον αντίθετό του ( $-x$ ), και θα εκτυπώνει τις τιμές που υπολόγισε.

##### Ζήτημα 2<sup>ο</sup>

Δημιουργήστε ένα πρόγραμμα που

- θα διαβάζει 3 ακεραίους αριθμούς ( $x, y, z$ ),
- θα υπολογίζει το άθροισμά τους ( $x+y+z$ ), το γινόμενο τους ( $x * y * z$ ),
- θα εκτυπώνει τους αριθμούς  $x, y, z$ , το άθροισμά τους και το γινόμενό τους.

##### Ζήτημα 3<sup>ο</sup>

Τι εκτυπώνουν τα ακόλουθα κομμάτια κώδικα;

```
int x=1;
x++;
++x;
--x;
x--;
cout << x;
```

```
int x=1, y;
y = x++;
y = ++x;
y = --x;
y = x--;
y = y + x--;
y = y + ++x;
cout << y;
```

```
int x=1, y=2, z=3, w=4, v=5;
cout <<
10*y - x + (2*z)/w - w/v*14/2;
```

Σκεφτείτε πρώτα την απάντησή σας και μετά επαληθεύστε δημιουργώντας τα αντίστοιχα προγράμματα. Σε κάθε γραμμή του κώδικα εισάγετε με σχόλια τις τιμές των αντίστοιχων μεταβλητών.

# 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]: 1.2, 1.3, 1.4, 1.5, Παράρτημα 1

[2]: Κεφ. 2, Παράρτημα Γ

- **Επιπλέον Ενασχόληση**

Δοκιμάστε να τρέξετε τα παραδείγματα των διαφανειών σε ολοκληρωμένα προγράμματα (με την συν/ση `main()`)