

343 – Εισαγωγή στον Προγραμματισμό

Τμήμα Μαθηματικών
Πανεπιστήμιο Ιωαννίνων

Ακαδημαϊκό Έτος 2015-2016

Χάρης Παπαδόπουλος
207δ, Β' όροφος
e-mail: 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

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

012 και
Εργαστ. Η'Υ, 1^ο ορόφου, Αναγνωστήριο
Εργαστ. Η'Υ, 1^ο ορόφου, Μικρό Ανασαστή

• Αρχική • Χρήσιμο Υλικό • Διαφάνειες • Εργαστήρια • Ημερολόγιο • Ανακοινώσεις

Περιληψη

Βασικά χαρακτηριστικά της γλώσσας προγραμματισμού C++. Σχεδίαση και ανάλυση υπολογιστικών προγραμμάτων, διάθωρα σταθμάτων, 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 αλγόριθμους και διαγράμματα ροής προγραμμάτων. Εφαρμογές σε προβλήματα αναζήτησης, ταξινόμησης και μαθηματικών προβλημάτων.

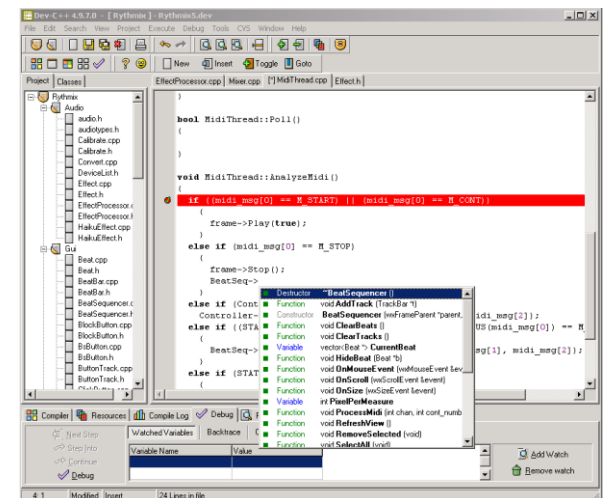
Στο μάθημα περιλαμβάνονται εργαστηριακές ασκήσεις στις οποίες η συμμετοχή είναι υποχρεωτική.

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. Χατζηγιαννάκης, 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^ο όροφο δίπλα από το Αναγνωστήριο
- Τα εργαστήρια θα ξεκινήσουν **Δευτέρα 19 Οκτωβρίου**

Εργαστήριο Δευτέρα (14:00-20:15)

A1 Δευτέρα 14:00-15:15	_____ - 10636
A2 Δευτέρα 15:15-16:30	10637 - 10702
A3 Δευτέρα 16:30-17:45	10703 - 10786
A4 Δευτέρα 17:45-19:00	10787 - 10881
A5 Δευτέρα 19:00-20:15	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 ^ο 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 ^ο 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 ^ο Εργαστήριο	
Πα, 13 Νοεμβρίου	Επανάληψη με Παραδείγματα	[1]: 5.1, 5.2, 5.4 [2]: Κεφ. 7
Δε, 16 Νοε	1 ^ο Quiz	
Πα, 20 Νοεμβρίου	Πίνακες (μονοδιάστατοι και πολυδιάστατοι)	[1]: 5.1, 5.2, 5.4 [2]: Κεφ. 7
Δε, 23 Νοε	4 ^ο 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 ^ο 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 ^ο 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, 19 Οκτ	1ο 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	2ο 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	3ο Εργαστήριο	
Πα, 13 Νοεμβρίου	Επανάληψη με Παραδείγματα	[1]: 5.1, 5.2, 5.4 [2]: Κεφ. 7
Δε, 16 Νοε	1ο Quiz	
Πα, 20 Νοεμβρίου	Πίνακες (μονοδιάστατοι και πολυδιάστατοι)	[1]: 5.1, 5.2, 5.4 [2]: Κεφ. 7
Δε, 23 Νοε	4ο 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	5ο 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	2ο 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 5

ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

Λογικές Εκφράσεις

- Μια έκφραση που μπορεί να είναι true (αληθής) ή false (ψευδής).
- Έκφραση: δύο εκφράσεις με αριθμούς ή μεταβλητές οι οποίοι συγκρίνονται με έναν **τελεστή σύγκρισης**
- **Τελεστές σύγκρισης:**

Σύμβολο	τελεστής	Κώδικας C++	Ερμηνεία	Παράδειγμα
=	==	x == y	true αν x=y, αλλιώς false	x+7==2*y
≠	!=	x != y	true αν x≠y, αλλιώς false	ans != 'n'
≤	<=	x <= y	true αν x≤y, αλλιώς false	count <= m+3
≥	>=	x >= y	true αν x≥y, αλλιώς false	a >=21
>	>	x > y	true αν x>y, αλλιώς false	time > limit
<	<	x < y	true αν x<y, αλλιώς false	x < y

Δημιουργία Λογικών Εκφράσεων – Τελεστής &&

- Συνδυασμός δυο λογικών εκφράσεων
- **Ερμηνεία &&**: Είναι αληθής όταν και οι δυο συγκρίσεις είναι αληθής, διαφορετικά είναι ψευδής.

Σύνταξη για λογική έκφραση με τον τελεστή &&

(Λογικη_Εκφραση_1) && (Λογικη_Εκφραση_2)

Παράδειγμα

```
if ( (score>0) && (score<10) )  
    cout << "η βαθμολογία είναι μεταξύ 0 και 10";  
else  
    cout << "η βαθμολογία δεν είναι μεταξύ 0 και 10";
```


Δημιουργία Λογικών Εκφράσεων – Τελεστής ||

- Συνδυασμός δυο λογικών εκφράσεων
- **Ερμηνεία ||**: Είναι αληθής όταν τουλάχιστον μια από τις δυο συγκρίσεις είναι αληθής, διαφορετικά είναι ψευδής.

Σύνταξη για λογική έκφραση με τον τελεστή ||

(Λογικη_Εκφραση_1) || (Λογικη_Εκφραση_2)

Παράδειγμα

```
if ( (x==1) || (x==y) )  
    cout << "Το x είναι 1 ή ίσο με y";  
else  
    cout << "Το x δεν είναι ούτε 1 ούτε y";
```

Συνεχόμενες ανισότητες

- Κλασική ανισότητα: $x < y < z$
δημιουργεί συντακτικό λάθος όταν γράφουμε:
 $x < y < z$
- Θα πρέπει: $(x < y) \ \&\& \ (y < z)$

Αποτίμηση λογικών εκφράσεων

- Αποτίμηση είναι μια λογική τιμή (true ή false)

```
bool result = (x < z) && (z < y) ;
```

- Πρώτα υπολογίζονται οι υποεκφράσεις

(x < z) και (z < y) ως true ή false

με κάποια συγκεκριμένη σειρά και στη συνέχεια ανάλογα με τους πίνακες αληθείας προκύπτει η συνολική τιμή

Εκφ1	Εκφ2	Εκφ1 && Εκφ2
true	true	true
true	false	false
false	true	false
false	false	false

Εκφ1	Εκφ2	Εκφ1 Εκφ2
true	true	true
true	false	true
false	true	true
false	false	false

Εκφ	!(Εκφ)
true	false
false	true

Παράδειγμα

```
! ( (y < 3) || (y > 7) )
```

- Έστω $y = 8$:

$(y < 3) \rightarrow \text{false}$

$(y > 7) \rightarrow \text{true}$

- Έπομένως:

`! (false || true)`

- Από πίνακα αληθείας: `false || true = true`

`! (true)`

- Από πίνακα αληθείας: `!true = false`

`false`

Κανόνες Προτεραιότητας Λογικών Εκφράσεων

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

1 αντιστροφή (not) : !

2 πράξεις συσχετισμού: <, >, <=, >=, ==

3 και : &&

4 ή : ||

```
(a < 90) && (b > 0.9) && (p == true)
```

⇔

```
a < 90 && b > 0.9 && p == true
```

Απλοποιημένη Αποτίμηση

$(x \geq 0) \ \&\& \ (y > 1)$

- Αν $x=-1$ τότε $(x \geq 0) \rightarrow \text{false}$

– Ανεξάρτητα από την τιμή του y , το συνολικό $\rightarrow \text{false}$

$(x \geq 0) \ || \ (y > 1)$

- Αν $x=1$ τότε $(x \geq 0) \rightarrow \text{true}$

– Ανεξάρτητα από την τιμή του y , το συνολικό $\rightarrow \text{true}$

- **Απλοποιημένη αποτίμηση:** στις δύο εκφράσεις μεταξύ των $\&\&$ ή $||$ η C++ αποτιμάει **πρώτα την αριστερή έκφραση** και στην συνέχεια ανάλογα με την τιμή προχωράει στην δεξιά.

```
if ( (kids!=0) && (pieces/kids) >=2 )  
    cout << "Κάθε παιδί μπορεί να πάρει 2 κομμάτια";
```

- Αν $kids=0$ τότε αποφεύγεται σφάλμα κατά την εκτέλεση

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 ()

Κανόνες Προτεραιότητας Γενικά

- 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 σχεσιακοί: <, >, <=, >=**
Υπολογίζονται από τα αριστερά προς τα δεξιά
- 6 ισότητας: ==, !=**
Υπολογίζονται από τα αριστερά προς τα δεξιά
- 7 λογικοί: &&, ||**
Υπολογίζονται από τα αριστερά προς τα δεξιά
- 8 εκχώρησης: =, +=, -=, *=, /=, %=**
Υπολογίζονται από δεξιά προς τα αριστερά.

Παράδειγμα

$$x + 1 > 2 \quad || \quad x + 1 < 3$$

$$(x + 1) > 2 \quad || \quad (x + 1) < 3$$

$$((x + 1) > 2) \quad || \quad ((x + 1) < 3)$$

Χρήση ακεραίων ως λογικές τιμές

- Ένας ακέραιος μπορεί να αναπαρασταθεί με μια τιμή `true` ή `false`, και αντίστροφα

Κάθε μη-μηδενικός ακέραιος \Leftrightarrow `true`

`0` \Leftrightarrow `false`

- Έστω:
`! time > limit` (όχι η τιμή `time` μεγαλύτερη από `limit`)
- Κανόνες προτεραιότητας:
`(! time) > limit`
- Αν `time=36`: `(! time) → false → 0`
- Δηλαδή `0 > limit → για limit=60 → false`
- Κανονικά όμως θα περιμέναμε `true`
- Για να αποφύγουμε τέτοιες καταστάσεις:
`!(time > limit)` ή `(time <= limit)`

Παραδείγματα Αποτίμησης

- Έστω `count = 0` και `limit = 10` :
 1. `count == 0 && limit < 20`
true
 2. `!(count == 12)`
true
 3. `(count == 1) && (x < y)`
false
 4. `(count < 10) || (x < y)`
true
 5. `!((count < 10) || (x < y)) && (count >= 0)`
false
 6. `(limit < 20) || ((limit/count) > 7)`
true
 7. `((limit/count) > 7) && (limit < 20)`
error
 8. `(5 && 7) + (!6)`
true \Leftrightarrow 1

Ενότητα 6

ΡΟΗ ΕΛΕΓΧΟΥ: IF-ELSE, SWITCH, ENUM

Η εντολή if-else

- Ανάλογα με μια συνθήκη που εξαρτάται από τις τιμές κάποιων σταθερών και μεταβλητών, εκτελούμε ορισμένες εντολές ή όχι.

Σύνταξη μιας απλής εντολής

```
if ( Λογική_Έκφραση )
    Εντολή_Κατάφασης;
else
    Εντολή_Άρνησης;
```

Παράδειγμα

```
if ( myscore > yourscore )
{
    cout << "Κέρδισα \n";
    total = total + 100;
}
else
{
    cout << "Έχασα \n";
    total = 0;
}
```

Σύνταξη ακολουθίας εντολών

```
if ( Λογική_Έκφραση )
{
    Εντολή_Κατάφασης_1;
    Εντολή_Κατάφασης_2;
    ...
}
else
{
    Εντολή_Άρνησης_1;
    Εντολή_Άρνησης_2;
    ...
}
```

Σύνθετες Εντολές

- Όταν θέλουμε να εκτελέσουμε ένα "μπλοκ εντολών" τότε περικλείουμε τις εντολές αυτές μέσα σε { }

```
if ( myscore > yourscore)
    cout << "Κέρδισα \n";
else
    cout << "Έχασα \n";
```

```
if ( myscore > yourscore)
{
    cout << "Κέρδισα \n";
}
else
{
    cout << "Έχασα \n";
}
```

```
if ( myscore > yourscore)
{
    cout << "Κέρδισα \n";
    total = total + 100;
}
else
{
    cout << "Έχασα \n";
    total = 0;
}
```

```
if ( myscore > yourscore) {
    cout << "Κέρδισα \n";
    total = total + 100;
} else {
    cout << "Έχασα \n";
    total = 0;
}
```

Κλασικό σφάλμα εκχώρησης

```
if ( x = 12 )  
    cout << "12 \n";  
else  
    cout << "!=12 \n";
```

- Δεν υπάρχει σφάλμα κατά τη διάρκεια μεταγλώττισης
- `(x = 12)` είναι εντολή εκχώρησης
- Αποθηκεύει την τιμή 12 στην x και επιστρέφει την τιμή 12
- Επομένως `(x = 12)` $\rightarrow 12 \rightarrow \neq 0 \rightarrow \text{true}$
- Δηλαδή **πάντα** εκτελείται η εντολή :
`cout << "12 \n";`
ανεξάρτητα από την τιμή του x
- Ένας τρόπος αντιμετώπισης: `12 = x` (μήνυμα σφάλματος)

Παραδείγματα

- Ο παρακάτω κώδικας οδηγεί σε διαίρεση με το 0;

```
j = -1;  
if ( (j > 0) && ( 1 / (j+1) > 10 ) )  
    cout << i << endl;
```

- Ποια είναι η έξοδος;

```
if (0)  
    cout << "Το 0 είναι true";  
else  
    cout << "Το 0 είναι false";  
cout << endl;
```

```
if (1)  
    cout << "Το 1 είναι true";  
else  
    cout << "Το 1 είναι false";  
cout << endl;
```


Παράλειψη της else

- Το κομμάτι της `else` είναι προαιρετικό

```
if ( sales >= minimum )  
    salary = salary + bonus;  
cout << "μισθός = € " << salary;
```

- Η εκτύπωση του μισθού γίνεται πάντα

Ένθετα if-else

- Οι δομές επιλογής μπορούν φυσικά να είναι και ένθετες πράγμα που μας επιτρέπει να έχουμε μεγαλύτερο έλεγχο πάνω στις εντολές που θα εκτελεστούν.

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

int main()
{
    float grade = 6.5 ;

    if( grade >= 5.0)
    {
        cout << "PASSED";
        if( grade >= 8.5)
            cout << "... with distinction!";
        cout << "\n";
    }
    else
        cout << "FAILED\n";

    return 0;
}
```

Παραδείγματα

- Ποια είναι η έξοδος του παρακάτω κώδικα;

```
int x = 2;
cout << "Ξεκίνα\n";
if( x <= 3 )
    if( x != 0)
        cout << "Γεια από τη δευτερη if\n";
    else
        cout << "Γεια από την else\n";
        cout << "Τελείωσε\n";
cout << "Ξεκίνα πάλι\n";
if( x > 3 )
    if( x != 0)
        cout << "Γεια από τη δευτερη if\n";
    else
        cout << "Γεια από την else\n";
        cout << "Τελείωσε πάλι\n";
```

```
int extra = 2; //-3, 0
if( extra < 0 )
    cout << "μικρό";
else if (extra == 0)
    cout << "μεσαίο";
else
    cout << "μεγάλο";
```

Παραδείγματα

- Ποια είναι η έξοδος του παρακάτω κώδικα;

```
int x = 2;
cout << "Ξεκίνα\n";
if( x <= 3 )
    if( x != 0)
        cout << "Γεια από τη δευτερη if\n";
    else
        cout << "Γεια από την else\n";
        cout << "Τελείωσε\n";
cout << "Ξεκίνα πάλι\n";
if( x > 3 )
    if( x != 0)
        cout << "Γεια από τη δευτερη if\n";
    else
        cout << "Γεια από την else\n";
        cout << "Τελείωσε πάλι\n";
```

```
int extra = 2; //-3, 0
if( extra < 0 )
    cout << "μικρό";
else if (extra == 0)
    cout << "μεσαίο";
else
    cout << "μεγάλο";
```

Η εντολή switch

Σύνταξη εντολής switch

```
switch ( Έκφραση_Ελέγχου )
{
    case Σταθερα_1:
        Ακολουθία_Εντολών_1;
        break;
    case Σταθερα_2:
        Ακολουθία_Εντολών_2;
        break;
    ...
    case Σταθερα_n:
        Ακολουθία_Εντολών_n;
        break;
    default:
        Προεπιλεγμένες_Εντολές;
}
```

Παράδειγμα

```
int vehicleClass;
double toll;
cout << "Δώσε τύπο οχήματος";
cin >> vehicleClass;
switch (vehicleClass)
{
    case 1:
        cout << "Επιβατικό";
        toll = 0.5;
        break;
    case 2:
        cout << "Λεωφορείο";
        toll = 1.5;
        break;
    case 3:
        cout << "Φορτηγό";
        toll = 2.0;
        break;
    default:
        cout << "Άγνωστος Τύπος";
}
```

Παράδειγμα με switch

```
char c;
bool afound=false, bfound=false;

cout << "give a character\n";
cin >> c;

switch (c)
{
    case 'a':
    case 'A':
        afound = true;
        break;
    case 'b':
    case 'B':
        bfound = true;
        break;
    default:
        break;
}
cout << "a's: " << afound << "b's: " << bfound;
```

αν δεν υπάρχει break τότε προχωράμε στην αμέσως επόμενη εντολή

Τύποι απαρίθμησης enum

- Λίστα δηλωμένων σταθερών (ακεραίων)

Σύνταξη εντολής switch

```
enum Μεταβλητή1 = { Λίστα_Σταθερών }
```

Παράδειγμα

```
enum monthlength = {JAN_LENGTH=31, FEB_LENGTH=28,  
MAR_LENGTH=31, APR_LENGTH=30, MAY_LENGTH=31, JUN_LENGTH=30,  
JUL_LENGTH=31, AUG_LENGTH=31, SEP_LENGTH=30, OCT_LENGTH=31,  
NOV_LENGTH=30, DEC_LENGTH=31};
```

```
enum Direction = {NORTH = 0, SOUTH = 1, EAST = 2, WEST = 3};
```

ισοδύναμο με

```
enum Direction = {NORTH, SOUTH, EAST, WEST};
```

```
enum myEnum= {ONE = 17, TWO, THREE, FOUR = -3, FIVE};
```

Αρχικές Τιμές: ONE = 17, TWO (18), THREE(19), FOUR(-3), FIVE(-2)

Παράδειγμα με enum

- Ποια είναι η έξοδος;

```
enum Direction { N, S, E, W }
```

```
cout << W << " " << E << " " << S << " " << N;
```

```
enum Direction { N = 5, S = 7, E = 1, W }
```

```
cout << W << " " << E << " " << S << " " << N;
```

- Το βλέπουμε ως έναν τύπο μεταβλητής

```
enum PetType { Cat, Dog, Bird };  
PetType p;  
//...  
switch(p)  
{  
    case Cat:    cout << "Hello cat\n";  
                break;  
    case Dog:    cout << "Hello dog\n";  
                break;  
    default:  
                break;  
}
```


Τριαδικός τελεστής συνθήκης

Σύνταξη τριαδικού τελεστή ? :

Λογική_Έκφραση ? τιμή_όταν_true : τιμή_όταν_false ;

Παράδειγμα

```
max = (n1 > n2) ? n1 : n2 ;
```

```
int main()
{
    float grade;
    cout << "Enter grade\n";
    cin >> grade;

    if( grade >= 5.0)
        cout << "Passed\n";
    else
        cout << "Failed\n";

    return 0;
}
```

```
int main()
{
    float grade;
    cout << "Enter grade\n";
    cin >> grade;

    cout << (grade >= 5.0 ?
            "Passed\n":
            "Failed\n");

    return 0;
}
```

Ενότητες 5 & 6

ΟΛΟΚΛΗΡΩΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

Ταξινόμηση μεταβλητής

- Γράψτε πρόγραμμα που ταξινομεί μια μεταβλητή `int n` σε μια από τις παρακάτω κατηγορίες και εμφανίζει ανάλογο μήνυμα: $n < 0$ ή $0 \leq n \leq 100$ ή $n > 100$

Ταξινόμηση μεταβλητής

- Γράψτε πρόγραμμα που ταξινομεί μια μεταβλητή `int n` σε μια από τις παρακάτω κατηγορίες και εμφανίζει ανάλογο μήνυμα: $n < 0$ ή $0 \leq n \leq 100$ ή $n > 100$

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter integer\n";
    cin >> n;

    if(n < 0)
        cout << n << " είναι μικρότερο του 0.\n";
    else if ((0 <= n) && (n <= 100))
        cout << n << " βρίσκεται μεταξύ 0-100.\n";
    else if (n > 100)
        cout << n << " είναι μεγαλύτερο του 100.\n";
    return 0;
}
```

Ταξινόμηση μεταβλητής

- Γράψτε πρόγραμμα που ταξινομεί μια μεταβλητή `int n` σε μια από τις παρακάτω κατηγορίες και εμφανίζει ανάλογο μήνυμα: $n < 0$ ή $0 \leq n \leq 100$ ή $n > 100$

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter integer\n";
    cin >> n;

    if(n < 0)
        cout << n << " είναι μικρότερο του 0.\n";
    else if ((0 <= n) && (n <= 100))
        cout << n << " βρίσκεται μεταξύ 0-100.\n";
    else if (n > 100)
        cout << n << " είναι μεγαλύτερο του 100.\n";
    return 0;
}
```

αν τα διαγράψουμε
πάλι έχουμε το ίδιο
αποτέλεσμα

Χαρακτηρισμός Βαθμολογίας

- Γράψτε πρόγραμμα που διαβάζει την βαθμολογία [0..100] και εκτυπώνει έναν ανάλογο χαρακτηρισμό:
 - 90 – 100: A
 - 80 – 89: B
 - 70 – 79: C
 - 60 – 69: D
 - < 59: F

Χαρακτηρισμός Βαθμολογίας

- Γράψτε πρόγραμμα που διαβάζει την βαθμολογία [0..100] και εκτυπώνει έναν ανάλογο χαρακτηρισμό:
 - 90 – 100: A
 - 80 – 89: B
 - 70 – 79: C
 - 60 – 69: D
 - < 59: F

```
#include <iostream>
using namespace std;
int main()
{
    int grade;
    cout << "Enter grade\n";
    cin >> grade;

    if ( grade >= 90 )
        cout << "A" ;
    else if ( grade >= 80 )
        cout << "B" ;
    else if ( grade >= 70 )
        cout << "C" ;
    else if ( grade >= 60 )
        cout << "D";
    else
        cout << "F" ;
}
```

Εμβαδόν και περιφέρεια κύκλου

- Γράψτε πρόγραμμα που διαβάζει την ακτίνα (R) ενός κύκλου και εκτυπώνει το εμβαδόν (πR^2) και την περιφέρεια ($2 \pi R$).

Εμβαδόν και περιφέρεια κύκλου

- Γράψτε πρόγραμμα που διαβάζει την ακτίνα (R) ενός κύκλου και εκτυπώνει το εμβαδόν (πR^2) και την περιφέρεια ($2 \pi R$).

```
#include <iostream>
using namespace std;
int main()
{
    double rad;
    cout << "Enter radius\n";
    cin >> rad;

    double area, circum;
    const double PI = 3.14;
    area = PI * rad * rad;
    circum = 2 * PI * rad;
    cout << area << " " << circum;
    return 0;
}
```

Εμβαδόν και περιφέρεια κύκλου

- Γράψτε πρόγραμμα που διαβάζει την ακτίνα (R) ενός κύκλου και εκτυπώνει το εμβαδόν (πR^2) και την περιφέρεια ($2 \pi R$).
- Θα πρέπει να ελέγξετε σωστά δεδομένα κατά την είσοδο:
 - δηλαδή αν η ακτίνα είναι θετικός αριθμός.
 - Σε αντίθετη περίπτωση πρέπει να εκτυπώνει αντίστοιχο μήνυμα λάθους και να τερματίζει το πρόγραμμα.

```
#include <iostream>
using namespace std;
int main()
{
    double rad;
    cout << "Enter radius\n";
    cin >> rad;

    double area, circum;
    const double PI = 3.14;
    area = PI * rad * rad;
    circum = 2 * PI * rad;
    cout << area << " " << circum;
    return 0;
}
```

Εμβαδόν και περιφέρεια κύκλου

- Γράψτε πρόγραμμα που διαβάζει την ακτίνα (R) ενός κύκλου και εκτυπώνει το εμβαδόν (πR^2) και την περιφέρεια ($2 \pi R$).
- Θα πρέπει να ελέγξετε σωστά δεδομένα κατά την είσοδο:
 - δηλαδή αν η ακτίνα είναι θετικός αριθμός.
 - Σε αντίθετη περίπτωση πρέπει να εκτυπώνει αντίστοιχο μήνυμα λάθους και να τερματίζει το πρόγραμμα.

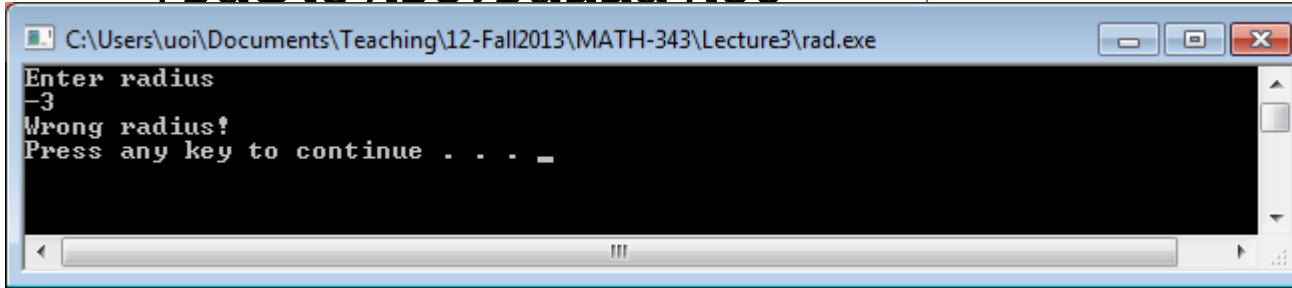
```
#include <iostream>
using namespace std;
int main()
{
    double rad;
    cout << "Enter radius\n";
    cin >> rad;

    if ( rad < 0 )
    {
        cout << "Wrong radius!" ;
        cout << endl;
        return 0;
    }

    double area, circum;
    const double PI = 3.14;
    area = PI * rad * rad;
    circum = 2 * PI * rad;
    cout << area << " " << circum;
    return 0;
}
```

Εμβαδόν και περιφέρεια κύκλου

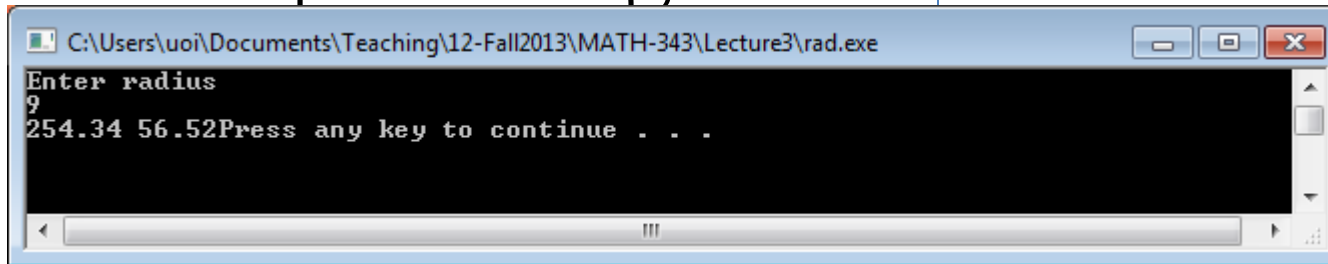
- Γράψτε πρόγραμμα που



```
C:\Users\uoι\Documents\Teaching\12-Fall2013\MATH-343\Lecture3\rad.exe
Enter radius
-3
Wrong radius!
Press any key to continue . . . . _
```

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

- Θα πρέπει να ελέγξετε



```
C:\Users\uoι\Documents\Teaching\12-Fall2013\MATH-343\Lecture3\rad.exe
Enter radius
9
254.34 56.52
Press any key to continue . . .
```

```
cin >> rad;
```

- Σε αντίθετη περίπτωση πρέπει να εκτυπώνει αντίστοιχο μήνυμα λάθους και να τερματίζει το πρόγραμμα.

```
    )
    "Wrong radius!" ;
endl;
return 0;
}
double area, circum;
const double PI = 3.14;
area = PI * rad * rad;
circum = 2 * PI * rad;
cout << area << " " << circum;
return 0;
}
```

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 → Lab-2.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

PreLab-2a.pdf

PreLab-2b.pdf

PreLab-2.pdf

- 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 → Lab-2.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-2.pdf

- ΠΡΙΝ το 2^ο Εργαστήριο θα πρέπει να ασχοληθείτε με τα (απλά) ζητήματα.
 - PreLab-2a.pdf (23/10)
 - PreLab-2b.pdf (30/10)

343 Εισαγωγή στον Προγραμματισμό :

ΠΡΟΕΤΟΙΜΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ & ΘΕΜΑΤΑ ΚΑΤΑΝΟΗΣΗΣ

2^ο Εργαστήριο – Α' μέρος

Χρήσιμο Υλικό:

- Βοηθητικό αρχείο: `readprintX.cpp` *διάβασμα ακεραίου και εκτύπωση του διπλάσιου ακεραίου*

Ζήτημα 1^ο

Δημιουργήστε ένα πρόγραμμα που θα διαβάζει τα μήκη a , b , c των τριών πλευρών ενός τριγώνου και θα εκτυπώνει το εμβαδόν του τριγώνου με βάση τον τύπο:

$$\sqrt{x(x-a)(x-b)(x-c)} \quad \text{όπου } x = \frac{a+b+c}{2}$$

Θα πρέπει να ελέγξετε για σωστά δεδομένα κατά την είσοδο: δηλαδή αν η τιμή μέσα στη ρίζα είναι θετικός αριθμός. Σε αντίθετη περίπτωση πρέπει να εκτυπώνει αντίστοιχο μήνυμα λάθους και να τερματίζει το πρόγραμμα.

Ζήτημα 2^ο

Δημιουργήστε ένα πρόγραμμα που θα διαβάζει έναν βαθμό $[0...100]$ (exam) από κάποιο μάθημα και θα εκτυπώνει τον χαρακτηρισμό του βαθμού: Δηλαδή

- αν $80 \leq \text{exam} \leq 100$ τότε ο χαρακτηρισμός είναι *άριστα*
- αν $65 \leq \text{exam} < 80$ τότε ο χαρακτηρισμός είναι *πολύ καλά*
- αν $50 \leq \text{exam} < 65$ τότε ο χαρακτηρισμός είναι *καλά*
- αν $\text{exam} < 50$ τότε ο χαρακτηρισμός είναι *αποτυχία*

Θα πρέπει να ελέγξετε για σωστά δεδομένα κατά την είσοδο: δηλαδή αν ο βαθμός (exam) ανήκει στο διάστημα $[0...100]$. Σε αντίθετη περίπτωση πρέπει να εκτυπώνει αντίστοιχο μήνυμα λάθους και να τερματίζει το πρόγραμμα.

Ζήτημα 3^ο

Δημιουργήστε ένα πρόγραμμα που θα διαβάζει έναν θετικό μονοψήφιο ακέραιο και θα εκτυπώνει τον ακέραιο αλφαριθμητικά (ένα, δύο, τρία, τέσσερα, ...) με χρήση της εντολής `switch`.

Θα πρέπει να ελέγξετε για σωστά δεδομένα κατά την είσοδο: δηλαδή αν ο ακέραιος είναι θετικός μονοψήφιος. Σε αντίθετη περίπτωση πρέπει να εκτυπώνει αντίστοιχο μήνυμα λάθους και να τερματίζει το πρόγραμμα.

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]: 2.1, 2.2, Παράρτημα 2

[2]: Κεφ. 4, Παράρτημα Α, ΣΤ

- **Επιπλέον Ενασχόληση**

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