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**

Τμήμα Μαθηματικών

343 Εισαγωγή στον Προγραμματισμό
Ακαδημαϊκό Έτος 2013-2014

Διδάσκων: Χάρης Παπαδόπουλος
Τηλ. Γρ.: 26510 – 08224
Γραφείο: 2076
E-mail: charis@cs.uoi.gr
Ώρες Γραφείου: Δευτέρα 11 - 13 και Παρασκευή 11 – 13
http: www.cs.uoi.gr/~charis/c343 ecourse.uoi.gr/course/view.php?id=xxxx
Ώρες Μαθημάτων: Διαλέξεις: **Πέμπτη 09-12** Εργαστήρια: **Τρίτη14:00-20:00**
Αίθουσα & Εργαστήρια: **012** και **Εργαστ. Η/Υ, 1ου ορόφου, Αναγνωστήριο** **Εργαστ. Η/Υ, 1ου ορόφου, Μικρό Ανσανσέρ**

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

Περίληψη

Βασικά χαρακτηριστικά της γλώσσας προγραμματισμού C++. Σχεδίαση και ανάλυση υπολογιστικών προγραμμάτων, διόρθωση σφαλμάτων, έλεγχος τεκμηρίωση, εγχειρίδιο χρήσης και αμυντικός προγραμματισμός. Βασικοί τύποι δεδομένων. Εντολές ελέγχου ροής προγράμματος. Είσοδος δεδομένω και έξοδος αποτελεσμάτων. Τύποι δεδομένων, συμβολοσειρές, και πίνακες.

Υποπρογράμματα, βασικές και αναδρομικές συναρτήσεις, διαβίβαση τιμών των παραμέτρων δια μέσω τιμής και δια μέσω διεύθυνσης. Διάρκεια (ζωής προσδιοριστών και κανόνες εμβέλειας και ορατότητας. Χρήση αρχείων. Δομές, εγγραφές. Λίστες και σύνθετοι τύποι δεδομένων. Προεπεξεργαστής κα βιβλιοθήκες. Τεχνικές αποτελεσματικού προγραμματισμού και εξοικείωση με αλγορίθμους και διαγράμματα ροής προγραμμάτων. Εφαρμογές α προβλήματα αναζήτησης, ταξινόμησης και μαθηματικών προβλημάτων.

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

Βιβλιογραφία

[1] W. Savitch. Πλήρης C++. Εκδόσεις Τζιόλα, 2011. Κωδικός Εύδ: 18548892

[2] H. Deitel and P. Deitel. C++ Προγραμματισμός 6η Εκδοση. Εκδόσεις Μ. Γκιούρδας, 2013. Κωδικός Εύδ: 12536819

[3] L. Jesse. Πλήρες εγχειρίδιο της C++. Εκδόσεις Α. Γκιούρδα, 2006. Κωδικός Εύδ: 12374

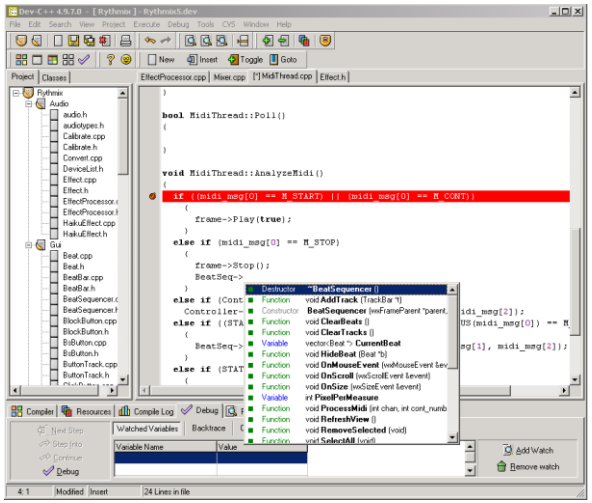
[4] Ν. Χατζηγιαννάκης. Η γλώσσα C++ σε βάθος. Εκδόσεις Κλειδάριθμος, 2008. Κωδικός Εύδ: 13761

Τρόπος Βαθμολόγησης

• Προϋπόθεση για την δυνατότητα βαθμολόγησης είναι η **επιτυχής παρακολούθηση** των υποχρεωτικών εργαστηρίων.
◦ Επιτυχής παρακολούθηση = μία (1) το πολύ απουσία στα **έξι (6)** εργαστηριακά μαθήματα

- Λογισμικό υλοποίησης προγραμμάτων : Bloodshed DevC++ :

  http://www.bloodshed.net/



Τμήματα Εργαστηρίων

- **Εργαστήριο (μεγάλο)**: Εργαστήριο Η/Υ 1ο όροφο δίπλα από το Αναγνωστήριο

- Τα εργαστήρια θα ξεκινήσουν **<u>Δευτέρα 19 Οκτωβρίου</u>**

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)

E: 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 E	20	21	22	**23 Θ**
26	27	28	29	**30 Θ**

Νοέμβριος 2015				
Δ	Τ	Τ	Π	Π
2 E	3	4	5	**6 Θ**
9 E	10	11	12	**13 Θ**
16 Q	17	18	19	**20 Θ**
23 E	24	25	26	**27 Θ**
30 E				

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	Ύλη βιβλιογραφίας
Πα, 9 Οκτωβρίου	**Εισαγωγικά μαθήματος & Δυαδική αναπαράσταση**	[1]: 1.1, Παράρτημα 3 [2]: Κεφ. 1, Β, Δ
Πα, 16 Οκτωβρίου	**Είσοδος/Έξοδος 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**	

Θ: διάλεξη (θεωρία)

E: 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 E	20	21	22	**23 Θ**
26	27	28	29	**30 Θ**

Νοέμβριος 2015				
Δ	Τ	Τ	Π	Π
2 E	3	4	5	**6 Θ**
9 E	10	11	12	**13 Θ**
16 Q	17	18	19	**20 Θ**
23 E	24	25	26	**27 Θ**
30 E				

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	Ύλη βιβλιογραφίας
Πα, 9 Οκτωβρίου	**Εισαγωγικά μαθήματος & Δυαδική αναπαράσταση**	[1]: 1.1, Παράρτημα 3 [2]: Κεφ. 1, Β, Δ
Πα, 16 Οκτωβρίου	**Είσοδος/Έξοδος 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 7

ΒΡΟΧΟΙ

Η εντολή while

- Εκτελεί κάποιες γραμμές κώδικα επαναληπτικά όσο μία συγκεκριμένη συνθήκη παραμένει αληθής
  - Μπορεί να επαναλαμβάνει μια εντολή ή πολλές
  - Αν η συνθήκη είναι αρχικά false, τότε δεν εκτελείται ποτέ

Σύνταξη while μιας εντολής

```
while ( Λογικη_Εκφραση )
    Εντολή;
```

Σύνταξη while με σώμα βρόχου πολλαπλών εντολών

```
while ( Λογικη_Εκφραση )
{
    Εντολή_1;
    Εντολή_2;
        ...
    Εντολή_n;
}
```

Παράδειγμα με while

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

int main()
{
    int count;
    cout << "Πόσους χαιρετισμούς θέλεις;";
    cin >> count;

    while( count > 0 )
    {
          cout << "Γεια σου ";
          count = count – 1;
    }

    cout << endl;
    cout << "Τελειώσαμε!\n";

    return 0;
}
```

Παράδειγμα1

```
Πόσους χαιρετισμούς θέλεις; 3
Γεια σου Γεια σου Γεια σου
Τελειώσαμε!
```

Παράδειγμα2

```
Πόσους χαιρετισμούς θέλεις; 0

Τελειώσαμε!
```

Η εντολή do-while

- Εκτελεί κάποιες γραμμές κώδικα επαναληπτικά όσο μία συγκεκριμένη συνθήκη παραμένει αληθής
  - Η συνθήκη επανάληψης ελέγχεται αφού εκτελεστεί το σώμα της επανάληψης ⇒ Εκτελείται τουλάχιστον μια φορά

Σύνταξη do-while μιας εντολής

```
do
    Εντολή;
while ( Λογικη_Εκφραση );
```

Σύνταξη do-while με σώμα βρόχου πολλαπλών εντολών

```
do
{
    Εντολή_1;
    Εντολή_2;
        ...
    Εντολή_n;
}
while ( Λογικη_Εκφραση );
```

Προσοχή στο ερωτηματικό στο τέλος, είναι υποχρεωτικό!!

Παράδειγμα με do-while

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

int main()
{
    int count;
    cout << "Πόσους χαιρετισμούς θέλεις;";
    cin >> count;

    do
    {
          cout << "Γεια σου ";
          count = count – 1;
    }
    while( count > 0 );

    cout << endl;
    cout << "Τελειώσαμε!\n";

    return 0;
}
```

Παράδειγμα1

```
Πόσους χαιρετισμούς θέλεις; 3
Γεια σου Γεια σου Γεια σου
Τελειώσαμε!
```

Παράδειγμα2

```
Πόσους χαιρετισμούς θέλεις; 0
Γεια σου
Τελειώσαμε!
```

Χρήση της do/while

- Χρησιμοποιείται συνήθως για σωστό διάβασμα εισόδου.
  - Κάθε φορά που διαβάζουμε μια λανθασμένη είσοδο ζητάμε από τον χρήστη νέα εισαγωγή δεδομένων:
  - Έστω ότι θα πρέπει exam ≥ 0:

```
cout << "Enter exam: ";
cin >> exam;

while (exam < 0)
{
    cout << "Enter exam: ";
    cin >> exam;
}
...
```

ενώ με do-while:

```
do
{
    cout << "Enter exam: ";
    cin >> exam;
}
while (exam < 0);
...
```

Χρήση της do/while

- Για σωστό *επαναληπτικό* διάβασμα θα πρέπει:
  Κάθε φορά που διαβάζουμε μια ***λανθασμένη*** είσοδο τότε ζητάμε από τον χρήστη νέα εισαγωγή δεδομένων.
- Π.χ.: Έστω ότι θα πρέπει $5 \leq exam \leq 10$:

```
( ( 5 <= exam ) && ( exam <= 10 ) )
```

```
do
{
    cout << "Enter exam: ";
    cin >> exam;
}
while ( (exam < 5) || (exam > 10) );
...
```

- Επομένως εάν θέλουμε να ισχύει μια συνθήκη τότε θα πρέπει στην συνθήκη της εντολής do - while( )να χρησιμοποιήσουμε την «αντίστροφη» συνθήκη.

Τελεστής αύξησης/μείωσης μέσα σε έκφραση

- Θυμηθείτε:

  Μέσα σε εκφράσεις

  – Επιστρέφουν την τιμή και μετά αλλάζουν τιμή `n++` ή `m--`
  – Αλλάζουν τιμή και μετά επιστρέφουν τιμή `++n` ή `--m`

- Επομένως σε μια έκφραση:

```
while ( count++ <= total )
{
        …
}
```

  – Πρώτα εξετάζουμε την συνθήκη και στη συνέχεια αυξάνουμε τον μετρητή
- Αν είχαμε `++count <= total`
  – Πρώτα αυξάνουμε τον μετρητή και στη συνέχεια εξετάζουμε την συνθήκη

Παράδειγμα αύξησης/μείωσης τελεστή

```
int main()
{
   int numItems, count, calItem, total;
   cout << "Πόσα κομμάτια έφαγες;";
   cin >> numItems;

   total = 0;
   count = 1;
   cout << "Δώσε θερμίδες/καθένα:\n";

   while( count++ <= numItems)
   {
         cin >> calItems;
         total = total + calItems;
   }

   cout << "Σύνολο θερμίδων:"
                << total << endl;
   return 0;
}
```

Παράδειγμα

```
Πόσα κομμάτια έφαγες; 7
Δώσε θερμίδες/καθένα:
300 60 1200 600 150 1 120
Σύνολο θερμίδων: 2431
```

Παραδείγματα

1.
```
int count = 3;
while (count-- > 0)
    cout << count << " ";
```

2.
```
int count = 3;
while (--count > 0)
    cout << count << " ";
```

3.
```
int n = 1;
do
    cout << n << " ";
while (n++ <= 3);
```

4.
```
int x = -42;
do
{
    cout << x << endl;
    x = x – 3;
}
while (x > 0);
```

5.
```
int x = 10;
do
{
    cout << x << endl;
    x = x – 3;
}
while (x > 0);
```

Η εντολή for

Σύνταξη εντολής for

```
for ( Απόδοση_αρχικών_τιμών; Λογικη_Εκφραση; Ενημέρωση)
    Εντολή_Σώματος;
```

Παράδειγμα

```
sum = 0;
for(n = 1; n <= 10; n++)
        sum = sum + n;
```

- Συνήθως χρησιμοποιείται όταν είναι προκαθορισμένο το πλήθος των επαναλήψεων
- Αν η Λογική_Έκφραση είναι αρχικά false, τότε οι εντολές δεν θα εκτελεστούν
- Η εντολή **`for`** μπορεί να γραφεί και σαν δομή **`while`**:

```
Απόδοση_αρχικών_τιμών;
    while (Λογικη_Εκφραση) {
                Εντολή_Σώματος;
                Ενημέρωση;
    }
```

Παραδείγματα με for

```
int number;
for( number = 100; number >= 0; number--)
        cout << number << " ";
```

```
int number;
for( number = 100; number >= 0; number--)
{
        cout << number << " ";
        number --;
}
```

- Ταυτόχρονη δήλωση μεταβλητής μέσα σε for:

```
for( int n = 1; n < 10; n++)
        cout << n << " ";
cout << n;
```

Λάθος!!

```
for( int n = 1; n < 10; n++)
        cout << n << " ";
```

```
int n;
for(n = 1; n < 10; n++)
        cout << n << " ";
cout << n;
```

Επισήμανση

- Μην χρησιμοποιείτε το ερωτηματικό στο τέλος της for

```
for( int n = 1; n < 10; n++);
        cout << " Γεια ";
```

- Θα εκτυπώσει μόνο μια φορά Γεια

- Κανονικά η εντολή

```
for( int n = 1; n < 10; n++);
```

λέει να εκτελέσεις 10 φορές την μηδενική (τίποτα) εντολή

- Επομένως η εντολή

```
cout << " Γεια ";
```

εκτελείται μετά τις 10 κενές επαναλήψεις, και ακριβώς μια φορά

Ατέρμονος βρόχος

- Τι εκτυπώνουν τα ακόλουθα προγράμματα;

```
#include <iostream>
using namespace std;
int main()
{
   int i = 0;

   while ( i < 10 )
      cout << i << endl;

   return 0;
}
```

```
int x = 2;

while ( x != 12 )
{
   cout << x << endl;
   x = x + 2;
}
```

```
int x = 1;

while ( x != 12 )
{
   cout << x << endl;
   x = x + 2;
}
```

Παραδείγματα

```
for( double sample=2; sample > 0; sample = sample -0.5)
        cout << sample << " ";
```

```
int n = 1024;
int log = 0;
for( int i = 1; i < n; i = i * 2)
     log++;
cout << log << endl;
```

```
int n = 1024;
int log = 0;
for( int i = 0; i < n; i = i * 2)
     log++;
cout << log << endl;
```

```
double total = 0.0;
for( int i = 1; i <= 10; i++)
     total = total + 1.0/i;
cout << total << endl;
```

Η εντολή break

- Τερματίζει την πλησιέστερη περικλείουσα εντολή βρόχου
  - Άμεση έξοδος από **`while`**, **`for`**, **`do`**/**`while`** or **`switch`**
  - Η εκτέλεση του προγράμματος συνεχίζεται με την επόμενη εντολή

```
int main()
{
   int count;

   for ( count = 1; count <= 10; count++)
   {
         if( count == 5 )
               break;
          cout << count << " ";
   }
   cout << "\nEnd!"<< endl;
   return 0;
}
```

Παράδειγμα

```
1 2 3 4
End!
```

Η εντολή continue

- Τερματίζει την τρέχουσα επανάληψη του βρόχου της πλησιέστερης περικλείουσας εντολής
  - Μεταφέρει τον έλεγχο στην ενημερωμένη έκφραση: η μεταβλητή ελέγχου ανανεώνεται αυτόματα

```
int main()
{
   int count;

   for ( count = 1; count <= 10; count++)
   {
         if( count == 5 )
               continue;
          cout << count << " ";
   }
   cout << "\nEnd!"<< endl;
   return 0;
}
```

Παράδειγμα

```
1 2 3 4 6 7 8 9 10
End!
```

Παράδειγμα break/continue

```
int main()
{
   int number, sum = 0, count = 1;
   cout << "Δώσε 4 αρνητικούς αριθμούς\n";

   while ( count <= 4 )
   {
         cin >> number;

         if( number >=0 )
         {
             cout << "ΣΦΑΛΜΑ: δόθηκε" <<
                "θετικός αριθμός"<< endl;
             break;
         }
         sum = sum + number;
         count++;
   }
   cout << "Το άθροισμα είναι "<< sum << endl;
   return 0;
}
```

Παράδειγμα

```
Δώσε 4 αρνητικούς αριθμούς
-1
-2
3
ΣΦΑΛΜΑ: δόθηκε θετικός αριθμός
Το άθροισμα είναι -3
```

Παράδειγμα break/continue

```
int main()
{
   int number, sum = 0, count = 1;
   cout << "Δώσε 4 αρνητικούς αριθμούς\n";

   while ( count <= 4 )
   {
         cin >> number;

         if( number >=0 )
         {
             cout << "ΣΦΑΛΜΑ: δόθηκε" <<
                "θετικός αριθμός"<< endl;
             continue;
         }
         sum = sum + number;
         count++;
   }
   cout << "Το άθροισμα είναι "<< sum << endl;
   return 0;
}
```

Παράδειγμα

```
Δώσε 4 αρνητικούς αριθμούς
-1
-2
3
ΣΦΑΛΜΑ: δόθηκε θετικός αριθμός
-3
-4
Το άθροισμα είναι -10
```

Φωλιασμένοι βρόχοι

- Πολλές φορές είναι χρήσιμο να ενσωματώσουμε βρόχο μέσα σε κάποιο άλλο βρόχο (διπλό loop)

```
int main()
{
   int i,j;

   for ( i = 0; i < 10; i++)
   {
      for ( j = 0; j < 5; j++)
         cout <<i<<","<<j<<" ";
   }
   cout << "\nEnd!"<< endl;
   return 0;
}
```

```
0,0 0,1 0,2 0,3 0,4 1,0 1,1 1,2 1,3 1,4 2,0 2,1 2,2 2,3 2,4 3,0 3,1 3,2 3,3 3,4
4,0 4,1 4,2 4,3 4,4 5,0 5,1 5,2 5,3 5,4 6,0 6,1 6,2 6,3 6,4 7,0 7,1 7,2 7,3 7,4
8,0 8,1 8,2 8,3 8,4 9,0 9,1 9,2 9,3 9,4
End!
Press any key to continue . . .
```

```
int main()
{
   int n,m;

   for ( n = 1; n <= 10; n++)
   {
      for ( m = 10; m >= 1; m--)
         cout << n <<" επί "<< m
            << " = " << n*m << endl;
   }
   cout << "\nEnd!"<< endl;
   return 0;
}
```

```
C:\Users\u...
1 epi 10 = 10
1 epi 9 = 9
1 epi 8 = 8
1 epi 7 = 7
1 epi 6 = 6
1 epi 5 = 5
1 epi 4 = 4
1 epi 3 = 3
1 epi 2 = 2
1 epi 1 = 1
2 epi 10 = 20
2 epi 9 = 18
2 epi 8 = 16
2 epi 7 = 14
```

Φωλιασμένοι βρόχοι

- Πολλές φορές είναι χρήσιμο να ενσωματώσουμε βρόχο μέσα σε κάποιο άλλο βρόχο (διπλό loop)

```
int main()
{
   int i,j;

   for ( i = 0; i < 10; i++)
   {
      for ( j = 0; j < 5; j++)
         cout <<i<<","<<j<<" ";
   }
   cout << "\nEnd!"<< endl;
   return 0;
}
```

Κλασικό (τυπογραφικό) λάθος:
Αν αλλάξουμε το j σε i τι θα γίνει;

Ενότητα 7

ΟΛΟΚΛΗΡΩΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

Κατάθεση

- Υπολογίστε την τιμή κάθε χρόνου μιας κατάθεσης €1000, με ετήσιο τόκο 5% χρησιμοποιώντας την σχέση

$$a = p\,(1 + r)^n$$

όπου:

p – αρχικό κεφάλαιο,
r – τόκος,
n – αριθμός χρόνων,
a – ποσό της κατάθεσης
μετά τον n χρόνο

- Υπολογίστε κάθε τιμή μέχρι 10 χρόνια

Κατάθεση

- Υπολογίστε την τιμή κάθε χρόνου μιας κατάθεσης €1000, με ετήσιο τόκο 5% χρησιμοποιώντας την σχέση

$$a = p\,(1 + r)^n$$

όπου:

p – αρχικό κεφάλαιο,
r – τόκος,
n – αριθμός χρόνων,
a – ποσό της κατάθεσης μετά τον n χρόνο

- Υπολογίστε κάθε τιμή μέχρι 10 χρόνια

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

int main()
{
   double a, p = 1000.0, r = 0.05;

   cout << "year \t amount\n";
   for(int i = 1; i <= 10; i++)
   {
       a = p * pow( (1.0 + r), i );
       cout << i << "\t" << a;
       cout << endl;
   }
   return 0;
}
```

Κατάθεση

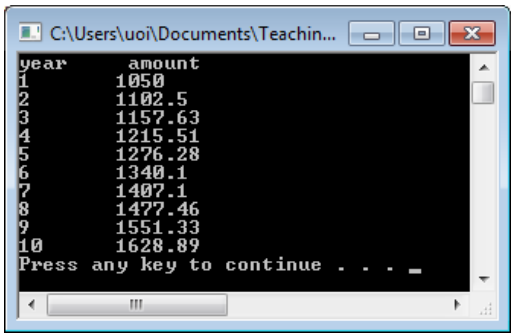
- Υπολογίστε την τιμή κάθε χρόνου μιας κατάθεσης €1000, με ετήσιο τόκο 5% χρησιμοποιώντας την σχέση

$$a = p\,(1 + r)^n$$

όπου:

p – αρχικό κεφάλαιο,
r – τόκος,
n – αριθμός χρόνων,
a – ποσό της κατάθεσης μετά τον n χρόνο

- Υπολογίστε κάθε τιμή μέχρι 10 χρόνια




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

int main()
{
   double a, p = 1000.0, r = 0.05;

   cout << "year \t amount\n";
   for(int i = 1; i <= 10; i++)
   {
       a = p * pow( (1.0 + r), i );
       cout << i << "\t" << a;
       cout << endl;
   }
   return 0;
}
```

Αριθμητική Πρόοδος

- Βρείτε το άθροισμα 1+2+...+10, δηλαδή το άθροισμα αριθμητικής προόδου

Αριθμητική Πρόοδος

- Βρείτε το άθροισμα 1+2+...+10, δηλαδή το άθροισμα αριθμητικής προόδου

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

int main()
{
   int n = 10, i, sum1, sum2;

   sum2 = n*(n+1)/2;
   sum1 = 0;
   for( i = 1; i <= n; i++ )
   {
       sum1 = sum1 + i;
       cout << i << "\t" << a;
       cout << endl;
   }
   cout << "Sum1: "<< sum1 << endl;
   cout << "Sum2: "<< sum2 << endl;
   return 0;
}
```

Παράδειγμα Εκτέλεσης

```
Sum1: 55
Sum2: 55
```

Μέσος όρος βαθμολογίας γνωστού πλήθους

- Μία τάξη 10 μαθητών έγραψε διαγώνισμα. Οι βαθμοί (ακέραιοι από 0 έως 100) του διαγωνίσματος είναι διαθέσιμοι σε σας. Βρείτε το μέσο όρο των βαθμών.

Μέσος όρος βαθμολογίας γνωστού πλήθους

```
int main()
{
   int exam, total;
   double avg;

   total = 0;
   for(int i = 1; i <= 10; i++)
   {
       do {
          cout << "Give exam ";
          cin >> exam;
       }
       while( (0 > exam) || (exam > 100) );
       total += exam;
   }
   avg = total/10.0;
   cout << "Average is: "<< avg << endl;
   return 0;
}
```

Μέσος όρος βαθμολογίας άγνωστου πλήθους

- Μία τάξη κάποιων μαθητών έγραψε διαγώνισμα. Οι βαθμοί (ακέραιοι από 0 έως 100) του διαγωνίσματος είναι διαθέσιμοι σε σας. Βρείτε το μέσο όρο των βαθμών.

Θεωρούμε ότι η τιμή -1 σταματάει την εκτέλεση

```
int main()
{
   int exam, total = 0, count = 0;
   double avg;

   cout << "Give exam (-1, to quit)";
   cin >> exam;

   while( exam != -1 )
   {
       total = total + exam;
       count = count + 1;

       cout << "Give exam (-1, to quit)";
       cin >> exam;
   }

   avg = total / count ;
   cout << "Average is: "<< avg << endl;



   return 0;
}
```

```
int main()
{
   int exam, total = 0, count = 0;
   double avg;

   cout << "Give exam (-1, to quit)";
   cin >> exam;

   while( exam != -1 )
   {
       total = total + exam;
       count = count + 1;

       cout << "Give exam (-1, to quit)";
       cin >> exam;
   }

   avg = total / count ;
   cout << "Average is: "<< avg << endl;



   return 0;
}
```

Σε κάθε διάβασμα της τιμής exam, θα μπορούσαμε:

```
do {
...
}
while(
 (exam<0||exam>100)
&&
 (exam != -1)
);
```

```
int main()
{
   int exam, total = 0, count = 0;
   double avg;

   cout << "Give exam (-1, to quit)";
   cin >> exam;

   while( exam != -1 )
   {
       total = total + exam;
       count = count + 1;

       cout << "Give exam (-1, to quit)";
       cin >> exam;
   }

   avg = total / count ;
   cout << "Average is: "<< avg << endl;



   return 0;
}
```

Προσοχή στη διαίρεση:
- το count μπορεί να είναι 0

```
int main()
{
   int exam, total = 0, count = 0;
   double avg;

   cout << "Give exam (-1, to quit)";
   cin >> exam;

   while( exam != -1 )
   {
       total = total + exam;
       count = count + 1;

       cout << "Give exam (-1, to quit)";
       cin >> exam;
   }
   if(count != 0) {
       avg = total / count ;
       cout << "Average is: "<< avg << endl;
   }
   else
       cout << "No input!" << endl;
   return 0;
}
```

Διαίρεση ακεραίων δίνει ακέραιο!!

Προσοχή, το
`static_cast<double>`
`(total /count)`
δεν δουλεύει

```
int main()
{
   int exam, total = 0, count = 0;
   double avg;

   cout << "Give exam (-1, to quit)";
   cin >> exam;

   while( exam != -1 )
   {
       total = total + exam;
       count = count + 1;

       cout << "Give exam (-1, to quit)";
       cin >> exam;
   }
   if(count != 0) {
       avg = total / static_cast<double>(count) ;
       cout << "Average is: "<< avg << endl;
   }
   else
       cout << "No input!" << endl;
   return 0;
}
```

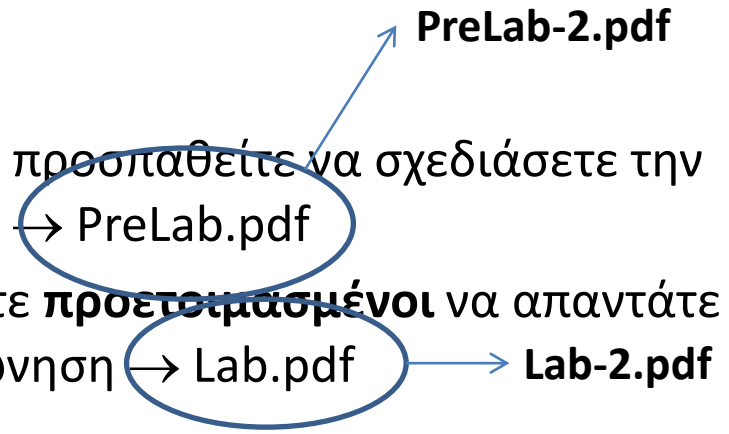
Ανάλυση αποτελεσμάτων

- Έχετε μία λίστα αποτελεσμάτων (1 = επιτυχία, 2 = αποτυχία) για μία εξέταση 10 μαθητών που μπορείτε να διαβάσετε. Γράψτε ένα πρόγραμμα που αναλύει τα αποτελέσματα: #επιτυχίες & #αποτυχίες.
- Επίσης εάν έχουν περάσει περισσότεροι των 8 μαθητών, τότε να τυπώνει το μήνυμα «πήγαμε πολύ καλά».

```
int main()
{
   int student = 1, passes = 0, failures = 0;
   int result;

   while( student <= 10 )
   {
       do {
             cout << "Give result (1=pass,2=fail)";
             cin >> result;
       }
       while( (result!= 1) && (result!= 2) );

       if(result==1)
              passes++;
       else
              failures++;

       student++;
   }
   cout << "Passed:" << passes << " Failed:"<< failures;
   if(passes >= 8)
       cout << "\nWell done! " << endl;
   return 0;
}
```

Εργαστήρια & Τεστ Quiz

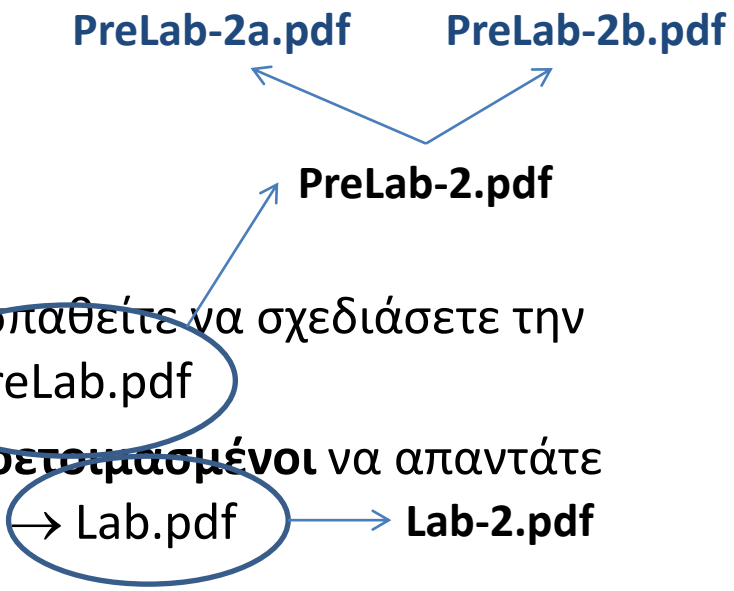
- Στα Εργαστήρια θα πρέπει:
  - Να κατανοείτε το πρόβλημα και να προσπαθείτε να σχεδιάσετε την λύση **πριν έλθετε στο εργαστήριο**. → PreLab.pdf
  - Να συμμετέχετε ενεργά και να είστε **προετοιμασμένοι** να απαντάτε σε ερωτήσεις σχετικές με την εκφώνηση → Lab.pdf



- 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
  - 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).

PreLab-2.pdf

- ΠΡΙΝ το 2ο 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

2ο Εργαστήριο – Α' μέρος

Χρήσιμο Υλικό:

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

Ζήτημα 1ο

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 2ο

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

- αν 80 ≤ exam ≤100 τότε ο χαρακτηρισμός είναι *άριστα*
- αν 65 ≤ exam < 80 τότε ο χαρακτηρισμός είναι *πολύ καλά*
- αν 50 ≤ exam < 65 τότε ο χαρακτηρισμός είναι *καλά*
- αν 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 3ο

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.

343 Εισαγωγή στον Προγραμματισμό :

ΠΡΟΕΤΟΙΜΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ & ΘΕΜΑΤΑ ΚΑΤΑΝΟΗΣΗΣ

2ο Εργαστήριο – Β' μέρος

Χρήσιμο Υλικό:

- Βοηθητικό αρχείο: sum10.cpp *διάβασμα 10 ακεραίων*

Ζήτημα 1ο

Δημιουργήστε ένα πρόγραμμα που να διαβάζει 8 πραγματικούς στο διάστημα [0 ... 1] και να υπολογίζει τον **μέγιστο** και τον **ελάχιστο** από τους 8 πραγματικούς αριθμούς. Αν ο αριθμός που διαβάζετε δεν είναι στο διάστημα [0 ... 1] τότε δεν λαμβάνεται υπόψη στην εύρεση του μεγίστου και ελαχίστου.

Ζήτημα 2ο

Ένα κολέγιο έχει μία λίστα αποτελεσμάτων (1 = επιτυχία, 2 = αποτυχία) για μία εξέταση αγνώστου πλήθους μαθητών. Γράψτε ένα πρόγραμμα που αναλύει τα αποτελέσματα: πόσοι είχαν επιτυχία, πόσοι είχαν αποτυχία καθώς επίσης και το ποσοστό της επιτυχίας και αποτυχίας.

Θα πρέπει να ελέγξετε για σωστά δεδομένα κατά την είσοδο: δηλαδή αν το αποτέλεσμα είναι 1 ή 2 ή κάποια άλλη τιμή που θα πρέπει να χρησιμοποιήσετε. Σε αντίθετη περίπτωση θα πρέπει να ζητάτε επαναληπτικά από τον χρήστη την σωστή τιμή.

Ζήτημα 3ο

Γράψτε ένα πρόγραμμα που εκτυπώνει ένα ορθογώνιο τρίγωνο από * («αστεράκια») με βάση και ύψος a. Την ποσότητα a τη δίνει ο χρήστης.

Παράδειγμα: Για a = 5, τότε εκτυπώνουμε:

```
*
**
***
****
*****
```

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]: 2.2, 2.3
[2]: Κεφ. 4, Κεφ. 5