

Η ΓΛΩΣΣΑ C

Μάθημα 1:

Το Πρώτο μας Πρόγραμμα σε C

Δημήτρης Ψούνης



www.psounis.gr

Περιεχόμενα Μαθήματος

A. Θεωρία

1. Κύκλος Ανάπτυξης Προγράμματος

1. Συγγραφή και Μεταγλώττιση ενός προγράμματος

2. Εκτέλεση του προγράμματος

2. Εγκατάσταση του Dev-C++

3. Το πρώτο πρόγραμμα σε C

1. Περιγραφή του προγράμματος

2. Συγγραφή του προγράμματος

3. Μεταγλώττιση του προγράμματος

4. Σύνδεση των αρχείων

5. Εκτέλεση του προγράμματος

6. Λίγα λόγια για το πρόγραμμα

B. Ασκήσεις

A. Θεωρία

1. Κύκλος Ανάπτυξης Προγράμματος

1. Συγγραφή και Μεταγλώττιση ενός προγράμματος

Πηγαίος κώδικας (source code) είναι η ακολουθία των εντολών σε μία γλώσσα προγραμματισμού, οι οποίες οδηγούν τον υπολογιστή να εκτελέσει τις επιθυμητές ενέργειες.

- Ο πηγαίος κώδικας μπορεί να γραφεί σε έναν **συντάκτη κειμένου (text editor)**, αν και συνήθως ο μεταγλωττιστής που έχουμε στην διάθεση μας διαθέτει και συντάκτη κειμένου.
- Στην C κάθε αρχείο πηγαίου κώδικα έχει κατάληξη **«.c»**.

Το αρχείο πηγαίου κώδικα που έχουμε γράψει πρέπει να **μεταγλωττιστεί (compile)** σε μια μορφή που αντιλαμβάνεται ο υπολογιστής.

- Έτσι κάθε **μεταγλωττιστής (compiler)** έχει μια επιλογή μεταγλώττισης που δέχεται σαν είσοδο το αρχείο πηγαίου κώδικα και παράγει σαν έξοδο το μεταγλωττισμένο αρχείο που ονομάζεται **αντικειμενικό αρχείο (object file)**.
- Το **αντικειμενικό αρχείο (object file)** έχει κατάληξη **«.o»**.

Συγγραφή Πηγαίου
Κώδικα:
Αρχείο «.c»

ΜΕΤΑΓΛΩΤΤΙΣΗ



Παραγωγή
Αντικειμενικού Αρχείου:
Αρχείο «.o»

A. Θεωρία

1. Κύκλος Ανάπτυξης Προγράμματος

2. Εκτέλεση του προγράμματος

Μετά την μεταγλώττιση ενός προγράμματος C σε αντικειμενικό αρχείο (object file) θα πρέπει να παράξουμε το **εκτελέσιμο αρχείο** (το αρχείο δηλαδή που μπορεί να τρέξει ως πρόγραμμα στον υπολογιστή).

- Η διαδικασία αυτή απαιτεί την **σύνδεση (linking)** του αντικειμενικού αρχείου με τις βιβλιοθήκες συναρτήσεων της C.
- Πάλι ένας μεταγλωττιστής έχει την επιλογή σύνδεσης που παράγει το εκτελέσιμο αρχείο (κατάληξη αρχείου **«.exe»**) το οποίο είναι το πρόγραμμα που θα τρέξει στον υπολογιστή.
- Οι περισσότεροι μεταγλωττιστές της C πραγματοποιούν τις διαδικασίες μεταγλώττισης και σύνδεσης αυτόματα (την δεύτερη αμέσως μετά την πρώτη) και συνεπώς παράγουν κατευθείαν το εκτελέσιμο αρχείο (κατάληξη **«.exe»**).

Παραγωγή
Αντικειμενικού Αρχείου:
Αρχείο «.o»

ΣΥΝΔΕΣΗ



Παραγωγή
Εκτελέσιμου Αρχείου:
Αρχείο «.exe»

A. Θεωρία

2. Εγκατάσταση του Dev-C++

1. Κατέβασμα και εγκατάσταση του Dev-C++

- Για την κατασκευή προγραμμάτων στα πλαίσια αυτής της σειράς μαθημάτων θα χρησιμοποιήσουμε τον μεταγλωττιστή Dev-C++.
- Στον ακόλουθο ιστοχώρο βρίσκουμε το αρχείο εγκατάστασης:

<http://sourceforge.net/projects/orwelldevcpp/files/Setup%20Releases/Dev-Cpp%205.5.3%20TDM-GCC%20x64%204.7.1%20Setup.exe/download>

- Στην συνέχεια εγκαθιστούμε το αρχείο που μόλις κατεβάσαμε (συμφωνούμε σε όλα).
- Για την αποθήκευση των προγραμμάτων τα οποία κατασκευάζουμε, είναι προτιμότερο να χρησιμοποιούμε κάποιον συγκεκριμένο φάκελο στον υπολογιστή μας (π.χ. έναν φάκελο programs στην Επιφάνεια Εργασίας).

A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

1. Περιγραφή του προγράμματος

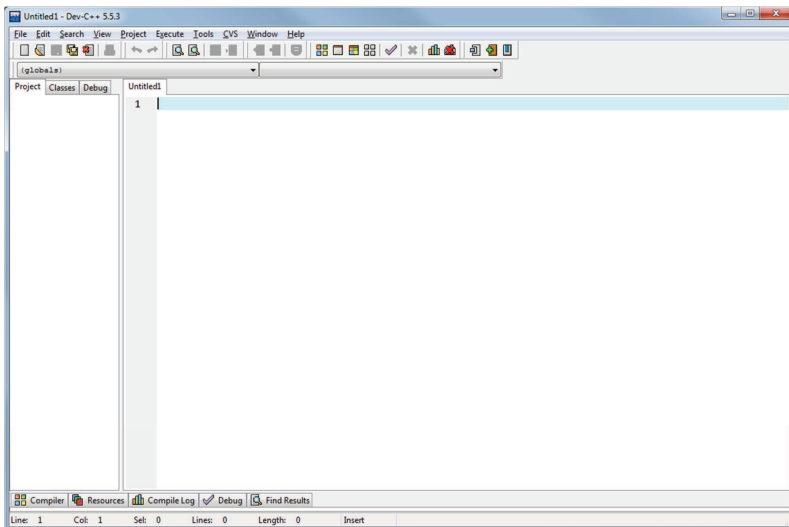
- Θα κατασκευάσουμε ένα πρόγραμμα που τυπώνει στην οθόνη το μήνυμα «Hello World!»
- Μέσω του προγράμματος αυτού θα δούμε τις ενέργειες που θα κάνουμε όταν θα καλούμαστε να γράψουμε ένα πρόγραμμα σε C, χρησιμοποιώντας το Dev C++.

A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

2. Συγγραφή του προγράμματος

- Ανοίγουμε το Dev-C++ και επιλέγουμε από το μενού File → New Source File



A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

2. Συγγραφή του προγράμματος

- Ο συντάκτης κειμένου είναι έτοιμος για να δεχθεί τις εντολές μας, γραμμένες σε γλώσσα C. Πληκτρολογούμε το πρόγραμμα μας ως ακολούθως:

```
#include <stdio.h>

int main() {
    printf("Hello World!\n");
    return 0;
}
```

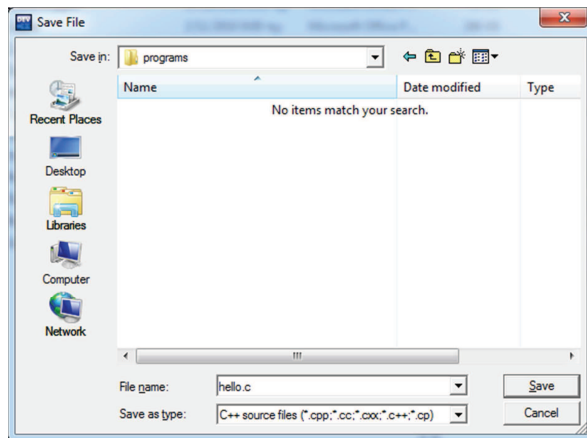
Σημείωση: Είναι σημαντικό ότι οτιδήποτε γράψουμε στο πρόγραμμα μας ακολουθεί τους συντακτικούς κανόνες της C (που θα μάθουμε αναλυτικά στα επόμενα μαθήματα). Έτσι αν κάτι δεν γράψουμε σωστά, τότε η διαδικασία της μεταγλώττισης θα αποτύχει!

A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

2. Συγγραφή του προγράμματος

- Έπειτα πρέπει να αποθηκεύσουμε το αρχείο στον υπολογιστή σε κάποιον φάκελο.
- Επιλέγουμε από το μενού File → Save As
- Στο παράθυρο που εμφανίζεται πληκτρολογούμε το όνομα του αρχείου ως «hello.c» και το αποθηκεύουμε στον φάκελο που επιθυμούμε:



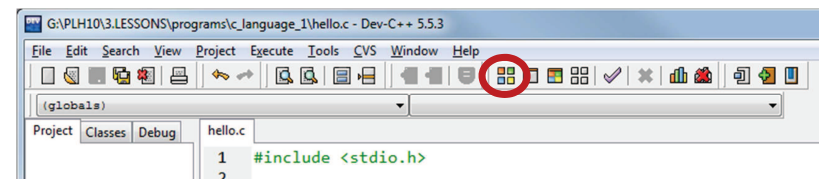
A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

3. Μεταγλώττιση του προγράμματος

- Αφού έχουμε συντάξει το πρόγραμμα πρέπει να το μεταγλωττίσουμε. Το Dev-C++ μας δίνει δύο τρόπους για τη μεταγλώττιση:

1. Είτε μέσω του κουμπιού compile



2. Είτε μέσω της επιλογής Execute → Compile του Μενου

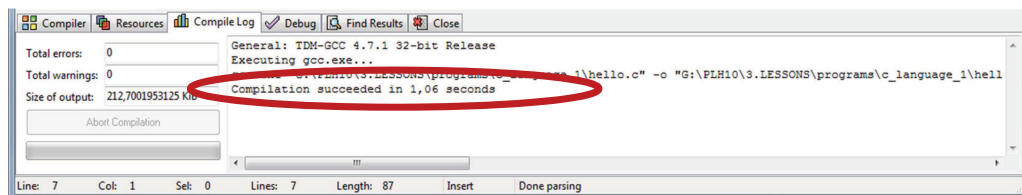
- Με οποιονδήποτε από τους παραπάνω δύο τρόπους κάνουμε μεταγλώττιση του πηγαίου κώδικα.

A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

3. Μεταγλώττιση του προγράμματος

- Αν όλα πάνε καλά, τότε εμφανίζεται ένα μήνυμα επιτυχίας στο κάτω μέρος της οθόνης:



- Παρατηρήστε επίσης ότι στον φάκελο στον οποίο έχουμε σώσει το πρόγραμμά μας, έχει εμφανιστεί και το αρχείο «hello.exe» που είναι το εκτελέσιμο αρχείο.

Σημείωση: Όταν το πρόγραμμα μας αποτελείται από μόνο ένα αρχείο, τότε αυτόματα γίνεται και μεταγλώττιση και σύνδεση, άρα παράγεται το τελικό εκτελέσιμο αρχείο.

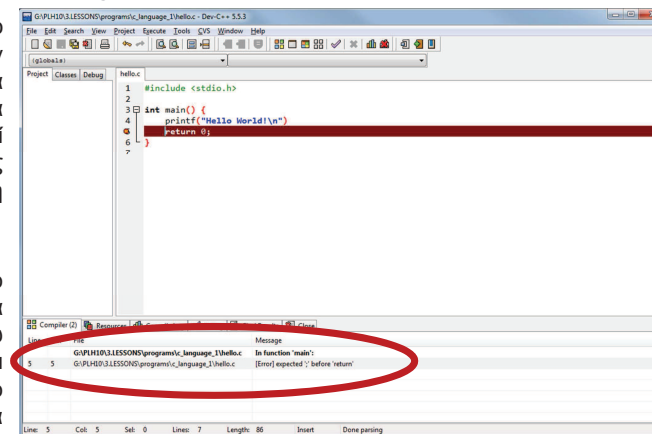
A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

3. Μεταγλώττιση του προγράμματος

- Αλλιώς αν έχουμε κάποιο συντακτικό λάθος (π.χ. αν ξεχάσαμε να βάλουμε ένα ερωτηματικό μετά από κάποια εντολή) τότε θα εμφανιστεί μήνυμα λάθους που θα μας καθοδηγήσει για την εύρεση του λάθους.

- Συγκεκριμένα κάνοντας διπλό κλικ σε ένα από τα μηνύματα που εμφανίζονται στο κάτω μέρος της οθόνης, εστιάζεται η γραμμή κώδικα που έχει το πρόβλημα και θα πρέπει να το διορθώσουμε μέχρι να ολοκληρωθεί επιτυχημένα η διαδικασία μεταγλώττισης.





A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

3. Μεταγλώττιση του προγράμματος

Λίγες Συμβουλές για το Μεταγλωττιστή:

- Όταν γράφουμε ολοκληρωμένα προγράμματα θα είναι πολύ συχνό να κάνουμε συντακτικά λάθη.
- Ο μεταγλωττιστής είναι αρκετά έξυπνος ώστε να αντιληφθεί συντακτικά λάθη, αλλά δεν είναι τόσο έξυπνος ώστε να μας καθοδηγήσει πλήρως στην εύρεση του λάθους.
- Παρατηρήστε για παράδειγμα ότι στο λάθος που κάναμε στο προηγούμενο παράδειγμα, το μήνυμα που βγάζει είναι ότι υπάρχει ένα συντακτικό λάθος πριν την εντολή return χωρίς να μας λέει ότι απλά λείπει ένα ερωτηματικό στην προηγούμενη γραμμή.
- Συνήθως όταν προκύπτουν λάθη κατά την μεταγλώττιση διορθώνουμε το πρώτο και έπειτα να επαναμεταγλωττίζουμε το πρόγραμμα.
- Αυτό γίνεται γιατί ένα συντακτικό λάθος μπορεί να οδηγήσει στην αλλοίωση της δομής του προγράμματος και ο μεταγλωττιστής να εμφανίσει περισσότερα λάθη από αυτά που πραγματικά υπάρχουν.

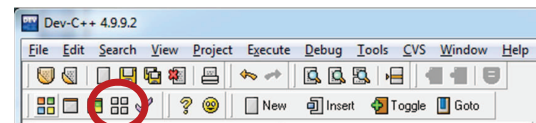


A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

4. Σύνδεση των Αρχείων

- Όπως αναφέρθηκε παραπάνω, όταν το πρόγραμμα αποτελείται από μόνο ένα αρχείο παράγεται αυτόματα το εκτελέσιμο αρχείο.
- Ωστόσο αν το πρόγραμμα μας αποτελείται από περισσότερα αρχεία θα παράγονται τα αντικειμενικά αρχεία πρώτα και έπειτα η **σύνδεση** θα γίνεται με το κουμπί Rebuild All

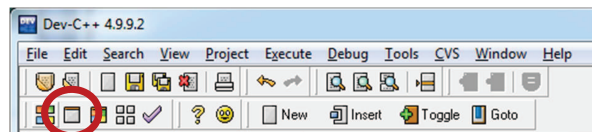


A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

5. Εκτέλεση του προγράμματος

- Η εκτέλεση του προγράμματος μπορεί να γίνει μέσω του κουμπιού Run:



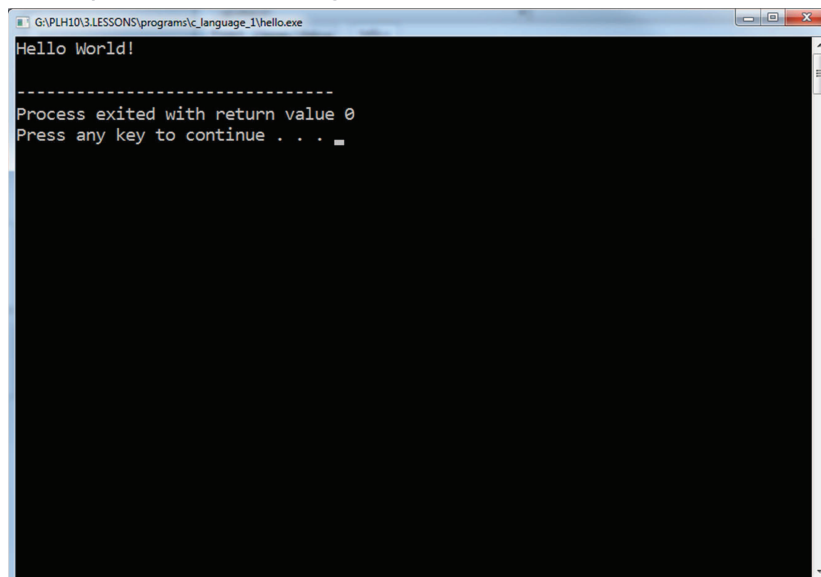
- ή ακόμη και κάνοντας διπλό κλικ στο αρχείο "hello.exe" που έχει δημιουργηθεί στον φάκελό μας.
- Η εκτέλεση του προγράμματος εμφανίζει την μαύρη οθόνη (κονσόλα) που έχει τυπωθεί το μήνυμα Hello World!



A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

5. Εκτέλεση του προγράμματος





A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

6. Λίγα Λόγια για το Πρόγραμμα

- Στα επόμενα μαθήματα θα μάθουμε αναλυτικά τα δομικά στοιχεία που χρησιμοποιήσαμε στο πρόγραμμα μας. Ας δούμε όμως κάποια πράγματα αρχικά:

- Η πρώτη γραμμή

```
#include <stdio.h>
```

- Προτρέπει τον μεταγλωττιστή να εισάγει την βιβλιοθήκη συναρτήσεων **stdio.h**.
- Στην βιβλιοθήκη αυτή έχει οριστεί η συνάρτηση **printf** που **τυπώνει** το όρισμά της στην **οθόνη**. Όταν χρησιμοποιούμε μία ήδη ορισμένη συνάρτηση πρέπει να ενσωματώνουμε (μέσω της οδηγίας #include) την βιβλιοθήκη στην οποία έχει οριστεί.
- Υπάρχουν πολλές βιβλιοθήκες συναρτήσεων που θα μάθουμε στην συνέχεια. Ενδεικτικά:
 - Η stdio.h έχει συναρτήσεις για είσοδο και έξοδο δεδομένων
 - Η stdlib.h έχει συναρτήσεις κυρίως για την διαχείριση της μνήμης
 - Η math.h έχει συναρτήσεις για την εκτέλεση μαθηματικών πράξεων



A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

6. Λίγα Λόγια για το Πρόγραμμα

- Η **συνάρτηση (function) main()**:

```
int main() {  
    printf("Hello World!\n");  
    return 0;  
}
```

- Είναι το σημείο αφετηρίας του προγράμματος.
- Κάθε πρόγραμμα που θα κατασκευάζουμε θα έχει **μία μοναδική συνάρτηση main()** και θα γνωρίζουμε ότι από εκεί ξεκινά η εκτέλεση ενός προγράμματος.
- Επίσης παρατηρήστε ότι οι εντολές της main είναι μέσα σε άγκιστρα. Τα άγκιστρα δείχνουν:
 - την αρχή του σώματος εντολών της main (με αριστερό άγκιστρο {)
 - το τέλος των εντολών της main (με δεξί άγκιστρο })



A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

6. Λίγα Λόγια για το Πρόγραμμα

- Η εντολή printf():

```
printf("Hello World!\n");
```

- Τυπώνει το όρισμά της στην οθόνη (κονσόλα).
- Εδώ η printf παίρνει σαν όρισμα μία συμβολοσειρά. Στην C οι συμβολοσειρές αναπαρίστανται μέσα σε διπλά εισαγωγικά.
- Το \n είναι ειδικός χαρακτήρας που αλλάζει γραμμή στην οθόνη. Ονομάζεται **χαρακτήρας αλλαγής γραμμής**.



A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

6. Λίγα Λόγια για το Πρόγραμμα

- Η εντολή return 0

```
return 0;
```

- η οποία μπαίνει πάντα στο τέλος της main επιστρέφει την τιμή 0 στο λειτουργικό σύστημα μετά την εκτέλεση της συνάρτησης.
- Περισσότερα για την εντολή αυτή θα αναφέρουμε σε επόμενα μαθήματα.



A. Θεωρία

3. Το πρώτο πρόγραμμα σε C

6. Λίγα Λόγια για το Πρόγραμμα

Το σημαντικότερο στον προγραμματισμό είναι ο πειραματισμός: να κάνουμε λάθη και να δούμε πως θα τα διορθώσουμε!

- Πειραματιστείτε με το πρόγραμμα αυτό, π.χ. εισάγοντας διαφορετικές εντολές printf και εισάγοντας το \n σε διαφορετικές θέσεις της συμβολοσειράς, ως εξής:

```
printf("\nHello \nWorld\n");
```



B. Ασκήσεις

Άσκηση 1

Κατασκευάστε ένα νέο πρόγραμμα (με όνομα radius.c) και πληκτρολογήστε το ακόλουθο πρόγραμμα:

```
#include <stdio.h>

int radius, area;

int main() {
    printf("Eisagete tin aktina:");
    scanf("%d",&radius);
    area=(int)3.1415 * radius * radius;
    printf("\n\nTo emvadon einai: %d\n", area);

    return 0;
}
```

Έπειτα εκτελέστε το πρόγραμμα με διαφορετικές εισόδους (Πειραματιστείτε!)



B. Ασκήσεις

Άσκηση 2

Κατασκευάστε ένα νέο πρόγραμμα (με όνομα triangle.c) και πληκτρολογήστε το ακόλουθο πρόγραμμα:

```
#include <stdio.h>

int i,j;

int main() {
    for (i=1; i<=8; i++) {
        for (j=1; j<=i; j++)
            printf("X");
        printf("\n");
    }

    return 0;
}
```

Παρατηρήστε τι παράγει το πρόγραμμα και προσπαθήστε να παράγετε ένα τρίγωνο 10 γραμμών τροποποιώντας κατάλληλα το πρόγραμμά σας.