

Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

Αρχές Προγραμματισμού Η/Υ (με τη γλώσσα C)

Διάλεξη #1

Παρασκευή, 5 Μαρτίου 2021

Παναγιώτης Παύλου

allos@mail.ntua.gr

Γενικά για το μάθημα

Εσείς και εμείς μαζί

Δομή Μαθήματος

Στο μάθημα υπάρχουν δύο παράλληλες ροές. Για τη ροή του προγραμματισμού με τη γλώσσα C, οι «εργαστηριακές διαλέξεις» πραγματοποιούνται τις Παρασκευές με τηλεδιδασκαλία λόγω των συνθηκών και σε **ένα ενιαίο τμήμα**, το πρωινό (9:00-10:45).

Σε κάθε διάλεξη θα δίνονται απλές **εργασίες** για το σπίτι και λίγες ημέρες αργότερα πρέπει να τις **υποβάλλετε ηλεκτρονικά**. Η βαθμολογία τους συμμετέχει στην τελική βαθμολογία του μαθήματος, βάσει όσων έχουν ειπωθεί με τον διδάσκοντα Επ.Καθηγητή κ. Λ. Αλεξόπουλο.

Η τελική εξέταση θα γίνεται πάνω σε Η/Υ με ηλεκτρονικό τρόπο, ανάλογα με τα τότε ισχύοντα.

Σκοπός

Ο σκοπός του προγραμματιστικού μέρους του μαθήματος, στο τέλος του εξαμήνου, είναι ο κάθε φοιτητής και η κάθε φοιτήτρια:

- Να έχει τη βασική αντίληψη για τα ζητήματα του προγραμματισμού
- Να μπορεί να κατανοήσει τη λειτουργία ενός προγράμματος ή ενός τμήματος κώδικα
- Να μπορεί ο κάθε συμμετέχων να δημιουργήσει ένα σχετικά απλό πρόγραμμα
- Να προετοιμαστεί για τη χρήση του προγραμματισμού στα υπόλοιπα μαθήματα της Σχολής

Υλικό

Το υλικό το οποίο σας παρέχεται στα πλαίσια του μαθήματος, σας παρέχει όλες τις απαιτούμενες γνώσεις ώστε να το ολοκληρώσετε επιτυχώς, ενώ θα είναι στη διάθεσή σας με διάφορες μορφές:

- Παρουσιάσεις από τις διαλέξεις (σε μορφή PDF)
- Ηλεκτρονική (σε PDF) και έντυπη μορφή του «Εγχειριδίου της Γλώσσας Προγραμματισμού C»
- Αναφορές (links) σε διάφορα site με σχετικό υλικό (ως links μέσα στα PDF)
- Προτεινόμενα σχετικά βιβλία διεθνούς βιβλιογραφίας
 - [K & R – The C programming language](#)
 - [Deitel-Deitel – C How to Program](#)
- Κοινότητες προγραμματιστών (ενδεικτικά το [SO](#) και [αυτό](#))
- Τη σελίδα ερωταποκρίσεων του μαθήματος

Τι θα μάθουμε

Η παραπάνω ύλη θα καλύπτει τουλάχιστον τα εξής αντικείμενα, με τη χρήση της γλώσσας προγραμματισμού C:

- Η πληροφορία στον Η/Υ
 - Αριθμοί
 - Πίνακες / Κείμενα
 - Δομές
- Αλγόριθμοι
 - Διαγράμματα ροής
 - Ροή εκτέλεσης
 - Έλεγχο ροής
- Συναρτήσεις
 - Δημιουργία
 - Βιβλιοθήκες συναρτήσεων
- Βασικές (standard) βιβλιοθήκες
 - Διαχείριση μνήμης
 - Ανάγνωση και εγγραφή αρχείων στον Η/Υ
 - Άλλες βιβλιοθήκες συναρτήσεων

Σημεία προσοχής

- Επειδή η περισσότερη ουσιαστική πληροφορία πάνω στον προγραμματισμό είναι στην Αγγλική γλώσσα, παρότι οι διαλέξεις είναι στα Ελληνικά, θα παρουσιάζονται και θα χρησιμοποιούνται οι **Αγγλικοί όροι**
- Ο προγραμματισμός είναι τέχνη γι' αυτό δεν γίνεται να τον μάθει κάποιος διαβάζοντας 1 εβδομάδα πριν τις εξετάσεις. **Απαιτεί παρακολούθηση, εξάσκηση και συζήτηση**
- **Ερωτήσεις** κατά τη διάρκεια του μαθήματος μπορούν να γίνουν σε συγκεκριμένα σημεία.
- Είμαστε **στη διάθεσή σας** για την επίλυση αποριών ή προβλημάτων με την προϋπόθεση ότι αυτή γίνεται αφού έχετε καταβάλει πρώτα τη σχετική προσπάθεια μόνοι σας και αφού έχετε ελέγξει μήπως υπάρχει ήδη απαντημένη η ερώτηση στο σχετικό site.

ΠΡΟΣΟΧΗ! Οι ερωτήσεις αποστέλλονται με e-mail στο:

c-programming@allos.gr

ενώ οι απαντήσεις θα δημοσιεύονται στο

<https://qna.c-programming.allos.gr>

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

<https://qna.c-programming.allos.gr>

- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLI ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



Γιατί γλώσσες προγραμματισμού

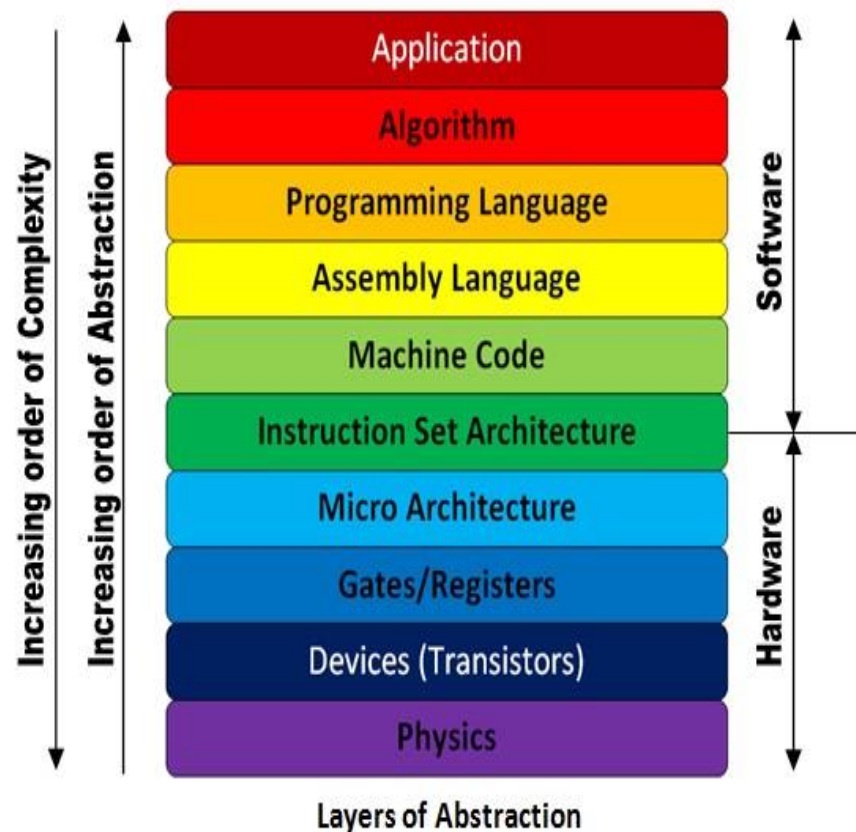
Τι ξεχωρίζει τη γλώσσα C

Αφαιρετικά επίπεδα λειτουργίας

Ο Η/Υ είναι μία συσκευή η οποία εκτελεί εντολές αδιάκοπα. Το λογισμικό ([software](#)) γράφεται για κάποιο **σκοπό**. Αποτελείται από τις εντολές που καθορίζουν το πώς θα λειτουργήσει η συσκευή ([hardware](#)), ώστε να επιτευχθεί ο παραπάνω σκοπός.

Τόσο το hardware όσο και το software παραδοσιακά στους Η/Υ οργανώνονται σε διάφορα επίπεδα αφαίρεσης ([abstraction layers](#)). Η λέξη επίπεδα (layers) δεν χρησιμοποιείται τυχαία αλλά για να τονίσει τη στρωματοποιημένη δομή τους.

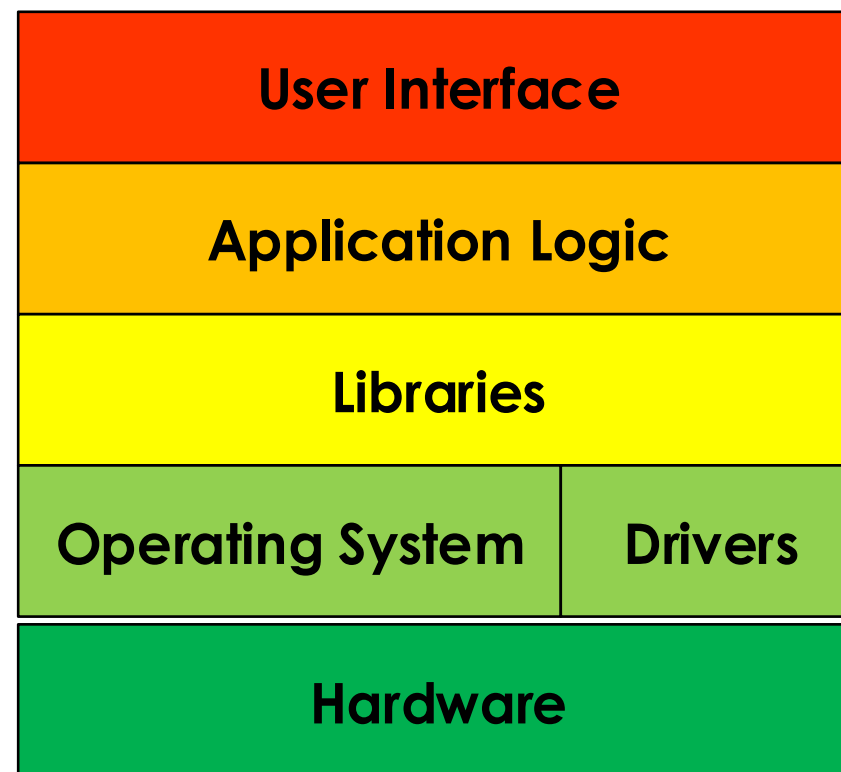
Πιο σημαντικό είναι όμως ο όρος «αφαίρεση» (abstraction) ο οποίος τονίζει ότι κάποιος μπορεί να ασχοληθεί με ένα μόνο layer γνωρίζοντας μόνο την εξωτερική συμπεριφορά των υπολοίπων layers στα οποία βασίζεται. Με τον ίδιο τρόπο που ο οδηγός ενός αυτοκινήτου δεν χρειάζεται να γνωρίζει μηχανολογία.



Αφαιρετικά επίπεδα λειτουργίας

Ως προς τη δομή του software:

- το υψηλότερο είναι η διεπαφή προς τον χρήστη (**user interface**) ή γενικότερα η διεπαφή του με το περιβάλλον
- το χαμηλότερο από αυτά τα επίπεδα είναι το λειτουργικό σύστημα (**operating system**) και οι οδηγοί (**drivers**) που επικοινωνούν απευθείας με το hardware
- όλα τα υπόλοιπα επίπεδα βρίσκονται ανάμεσα σε άλλα επίπεδα αφαίρεσης



Παραδείγματα



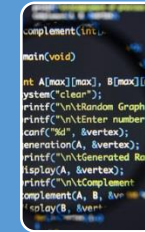
http

- Web σελίδα από τον server στον browser



Αρχεία και φάκελοι

- Το λογισμικό και οι χρήστες ασχολούνται με αρχεία και φακέλους



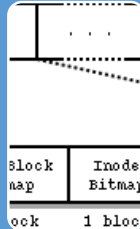
Γλώσσα υψηλού επιπέδου

- Κώδικας σε C



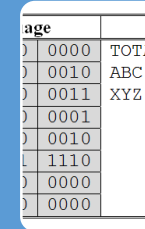
TCP

- Μεταφορά πληροφορίας από Η/Υ σε Η/Υ



Filesystem

- Η δομή της αποθήκευσης των αρχείων στον δίσκο



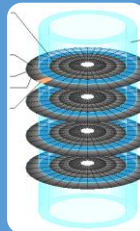
Γλώσσα Μηχανής

- Εντολές «επιπέδου επεξεργαστή»



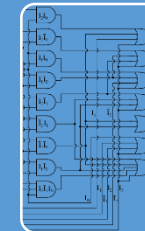
Ethernet

- Αφορά τη φυσική σύνδεση Η/Υ με Η/Υ



Blocks

- Η τμηματοποίηση του δίσκου, ανάλογα με την τεχνολογία του



Λογικά Κυκλώματα

- Λογικές πύλες, Transistors

Γλώσσες προγραμματισμού

Υπάρχουν πάρα πολλές γλώσσες προγραμματισμού.
Κάποιες από αυτές βασίζονται σε άλλες προγενέστερες.
Κάποιες από αυτές είναι κατάλληλες για συγκεκριμένες
εργασίες. Πασίγνωστες γλώσσες προγραμματισμού είναι οι:

C/C++	Java	JavaScript	Python	FORTRAN
C#	Kotlin	Perl	Dart	Assembly
VisualBasic	Lua	Pascal	PHP	Go

Αυτές που αναφέρονται με κόκκινα γράμματα είναι γλώσσες που έχουν παρόμοιο συντακτικό με τη C. Αυτό σημαίνει ότι μαθαίνοντας μία από αυτές γνωρίζουμε το βασικό συντακτικό και για τις υπόλοιπες.

Άλλες γλώσσες του Η/Υ

Υπάρχουν επίσης γλώσσες για τον υπολογιστή οι οποίες δεν αφορούν εντολές προς τον επεξεργαστή, άρα δεν αποτελούν γλώσσες προγραμματισμού. Τέτοιες είναι οι:

- HTML, XML και JSON, οι οποίες περιγράφουν δεδομένα
- CSS, που περιγράφει την εμφάνιση δεδομένων (κυρίως HTML)
- XSLT, που περιγράφει μετασχηματισμούς δεδομένων και πολλές άλλες.

Γλώσσες που ξεχωρίζουν

Ιδιαίτερη σημασία πρέπει να δοθεί στη γλώσσα Assembly η οποία είναι μία «πιο εμφανίσιμη» μορφή της γλώσσας μηχανής. Αυτή είναι η γλώσσα που «μιλάει» ο ίδιος ο επεξεργαστής.

Οποιοσδήποτε κώδικας εκτελείται σε έναν Η/Υ είναι οπωσδήποτε σε γλώσσα μηχανής.

Η γλώσσα μηχανής, άρα και η Assembly είναι **διαφορετική** για κάθε τύπο επεξεργαστή. Αυτό καθιστά αρκετά περίπλοκο το να γνωρίζει κάποιος όλες τις **παραλλαγές**. Επίσης δημιουργεί μεγάλο πρόβλημα στο να δημιουργηθεί ένας **ενιαίος** κώδικας που να λειτουργεί σε διαφορετικούς επεξεργαστές και λειτουργικά συστήματα. Να είναι όπως λέμε cross-platform.

Τη λύση στα παραπάνω προβλήματα, παραμένοντας ταυτόχρονα όσο το δυνατόν πιο κοντά στη λογική του επεξεργαστή, μας τη δίνει η C.

Γιατί η C?

Η C είναι μία γλώσσα που λίγοι προγραμματιστές δεν θα την μάθουν στη ζωή τους. Μερικά σημαντικά χαρακτηριστικά της είναι:

- Είναι πολύ κοντά στην γλώσσα μηχανής, παρέχει τις ίδιες λειτουργίες
- Αυτό την καθιστά διαχρονική γλώσσα, δεν αναμένεται να εγκαταληφθεί
- Πάνω σε αυτή βασίζονται πολλές σύγχρονες γλώσσες (με πιο γνωστές τις Java, JavaScript, PHP)
- Ένας κώδικας σε C που χρησιμοποιεί τις τυπικές βιβλιοθήκες αναμένεται να λειτουργεί απροβλημάτιστα σε οποιοδήποτε λειτουργικό σύστημα, εφόσον γίνει compile γι' αυτό. Είναι δηλαδή cross-platform.
- Λόγω της ευρείας χρήσης της έχει αρκετές έτοιμες βιβλιοθήκες πέρα από τις τυπικές

Υψηλού επιπέδου γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού (εκτός της γλώσσας μηχανής και της Assembly) ονομάζονται και υψηλού επιπέδου γλώσσες με την έννοια ότι αποτελούν ένα abstraction layer **πάνω** από τη γλώσσα μηχανής.

Ο κώδικας σε αυτές τις γλώσσες γράφεται σε απλά αρχεία κειμένου που μπορούν να ανοίξουν ακόμα και με το σημειωματάριο (notepad) των Windows.

Για να εκτελεστούν όμως πρέπει με κάποιον τρόπο να μετατραπούν σε γλώσσα μηχανής. Σε αυτό υπάρχουν τρεις λύσεις που χρησιμοποιούνται:

1. [Interpreter](#) (διερμηνέας)
Φορητότητα (Portability) μεταξύ συστημάτων, Προϋποθέτει την εγκατάσταση του στο σύστημα που εκτελείται ο κώδικας – Scripting γλώσσες : JavaScript, Python, PHP, Perl, Lua, Dart
2. Compiler (μεταφραστής)
Ταχύτητα εκτέλεσης, ο τελικός κώδικας είναι αυτόνομος – Compiled γλώσσες : C/C++, FORTRAN, Pascal, Go
3. [JIT/AOT Compiler](#) (Just-in-time/Ahead-of-time compiler)
Μίξη των παραπάνω – Compiled σε bytecode : VisualBasic, C#, Java, Kotlin

Παράδειγμα Scripting

Μία πολύ διαδεδομένη **scripting** γλώσσα είναι η PHP. Ακολουθεί ένα πρόγραμμα που εμφανίζει έναν χαιρετισμό στην οθόνη:

```
<?php
```

Edit

```
echo 'Hello world!';
```

```
sleep(10);
```

Το οποίο εκτελείται (εφόσον είναι εγκατεστημένη η PHP στον υπολογιστή) γράφοντας στη γραμμή εντολών:

```
> php file-name.php
```

Run

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

<https://qna.c-programming.allos.gr>

- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLion ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!



Η γλώσσα C

Πρώτα στοιχεία της και περιβάλλον ανάπτυξης (IDE)

Το 1^ο πρόγραμμα σε C

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

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

Δίπλα παρουσιάζεται το αντίστοιχο πρόγραμμα με αυτό της PHP, γραμμένο στη γλώσσα C.

Είναι και πάλι ένα απλό αρχείου κειμένου. Για να εκτελεστεί πρέπει να μετατραπεί σε εκτελέσιμο αρχείο.

Τη διαδικασία αυτή την αποκαλούμε συνήθως compilation, όμως αυτό είναι ανακριβές. Το compilation είναι ένα (το κυριότερο) βήμα της. Όλη η διαδικασία ονομάζεται building και περιλαμβάνει μεταξύ άλλων και το βήμα του linking. Το linking συνδέει τον compiled κώδικα που έχει γραφεί, με όλο τον βοηθητικό κώδικα που απαιτείται για την εκτέλεση του προγράμματος.

IDE : Το περιβάλλον ανάπτυξης

Στη C, αλλά και στις περισσότερες υψηλού επιπέδου γλώσσες (χωρίς να είναι αυτό απαραίτητο), όλη η διαδικασία της ανάπτυξης ενός προγράμματος, πραγματοποιείται μέσα σε ένα λογισμικό το οποίο ονομάζεται Integrated Development Environment (δηλαδή Ολοκληρωμένο Περιβάλλον Ανάπτυξης).

Αυτό περιλαμβάνει τουλάχιστον:

1. τον **editor** (τον επεξεργαστή κειμένου) όπου γράφεται ο κώδικας
2. τον **builder** όπου γίνεται η μετατροπή του κώδικα σε εκτελέσιμο
3. τη γραμμή εντολών όπου γίνεται η εκτέλεση του κώδικα
4. τον **debugger**, όπου βοηθά τον εντοπισμό και την επίλυση σφαλμάτων (bugs) στον κώδικα

Διαθέσιμα IDEs

Για τη C υπάρχουν πολλά διαθέσιμα IDEs. Μερικά από αυτά είναι:

- JetBrains CLion
- Bloodshed C++
- Netbeans
- Codeblocks
- Eclipse
- Microsoft Visual Studio
- VSCode

Από αυτά, στο μάθημα θα

χρησιμοποιούμε το **CLion**, το οποίο είναι ένα επαγγελματικό IDE.

Υπάρχει ήδη εγκατεστημένο στο PC-Lab της Σχολής, αλλά μπορείτε (και πρέπει) να το εγκαταστήσετε σε έναν υπολογιστή σας ώστε να μπορείτε να κάνετε τις εργασίες σας και από το σπίτι.

CLion : Εγκατάσταση και 1^η χρήση

Η εγκατάσταση του CLion γίνεται με τα ακόλουθα βήματα:

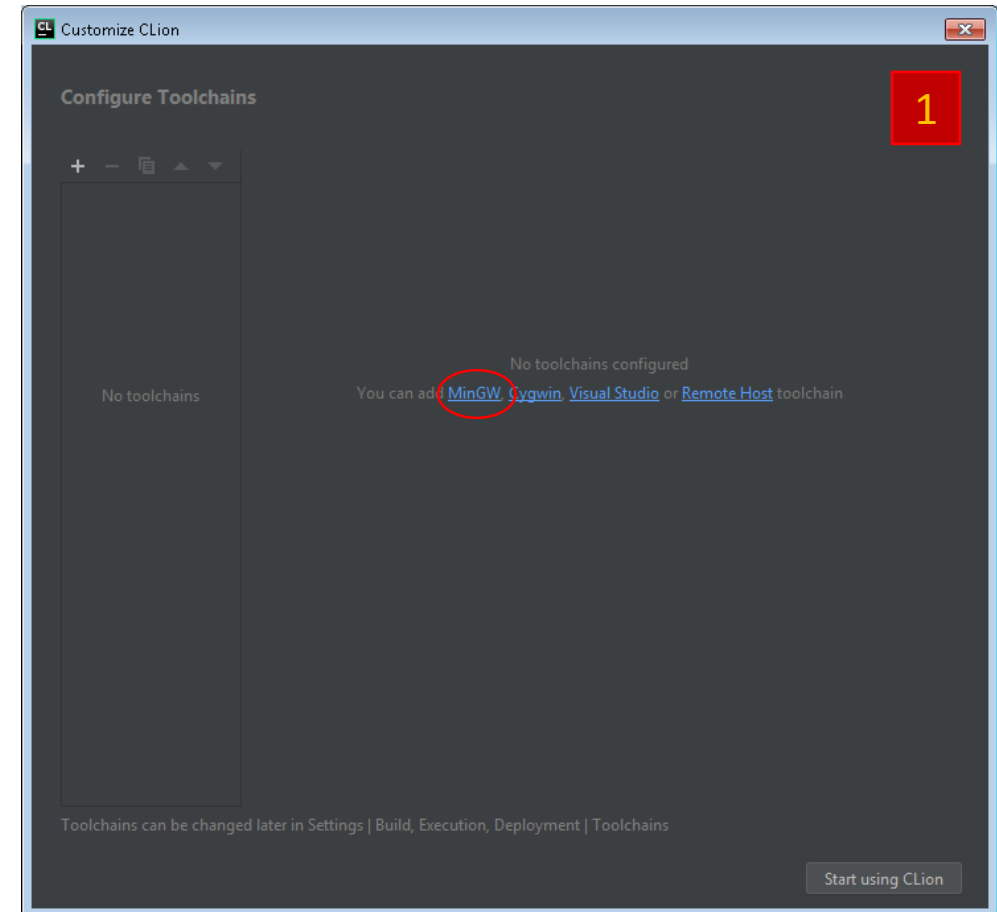
1. [Δημιουργία](#) κωδικού χρήστη στην JetBrains, χρησιμοποιώντας το e-mail (mc#####@mail.ntua.gr) του ΕΜΠ
2. [Λήψη](#) και εγκατάσταση
3. [Ρύθμιση](#) κατά την πρώτη χρήση (δεν απαιτούνται κάποια plugins)
4. [Ενεργοποίηση](#) του στον υπολογιστή σας (επιτρέπεται μόνο ένας Η/Υ ανά φοιτητή)
5. Ρύθμιση του toolchain για CMake με το MinGW, δηλαδή το εργαλείο με το οποίο γίνεται το build

CLion : Ρύθμιση toolchain & MinGW (1/4)

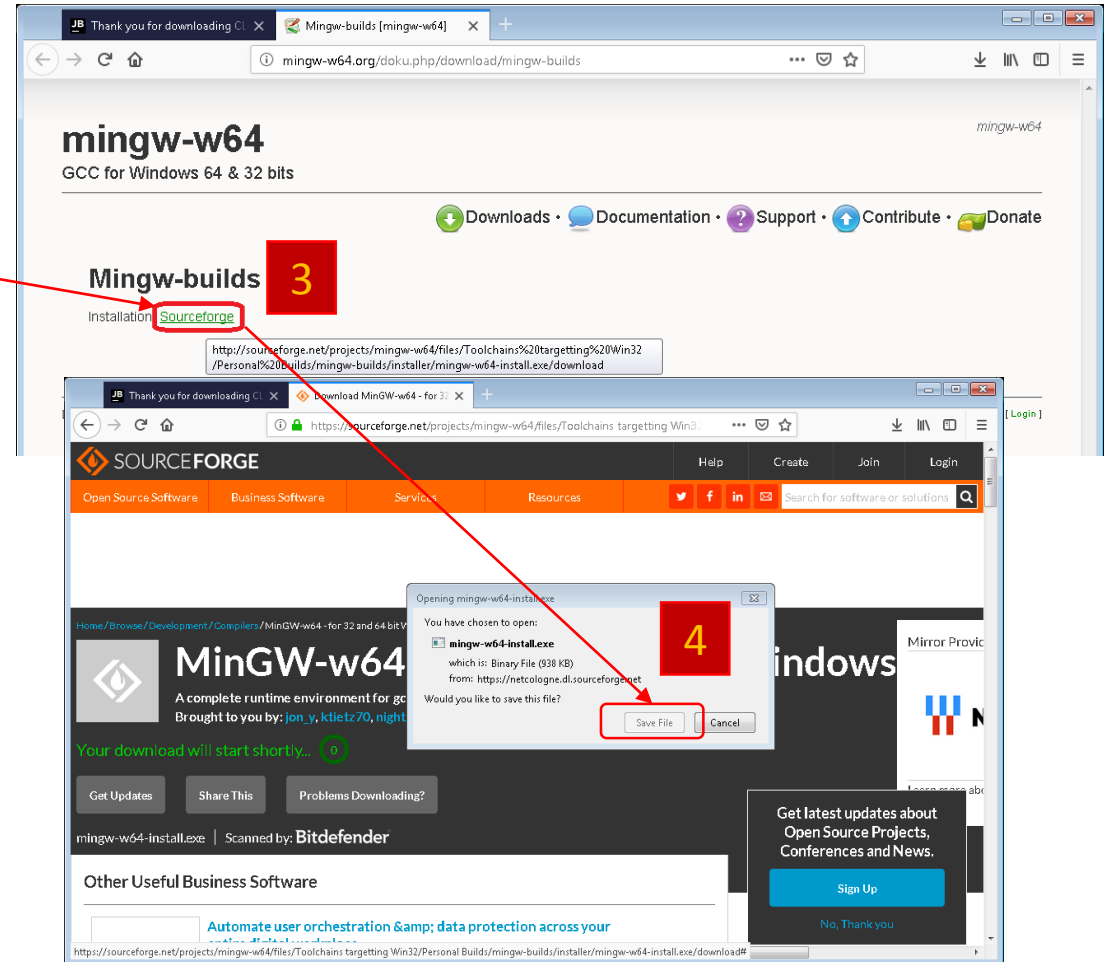
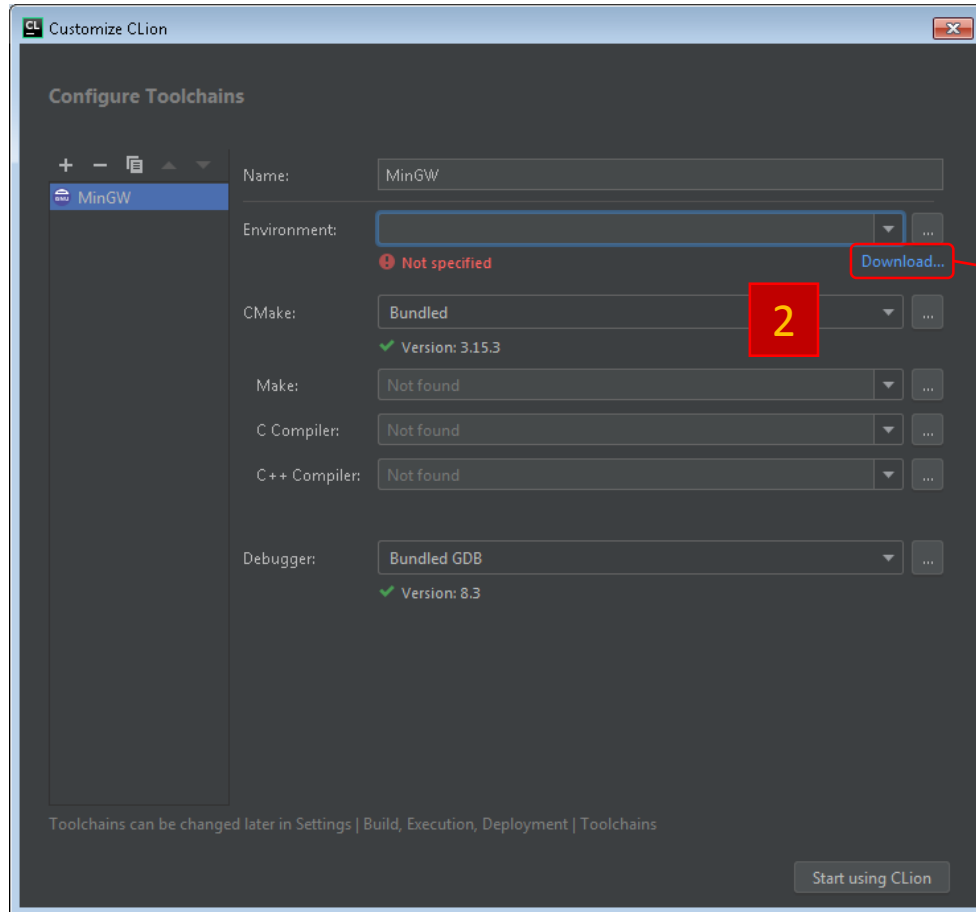
Την πρώτη φορά που θα εκτελέσετε το CLion θα πρέπει να του υποδείξετε τη θέση του compiler. Αυτό ακόμα δεν θα το έχετε εγκαταστήσει. Κατά την 1^η εκτέλεση εμφανίζεται το παρακάτω παράθυρο για να οριστεί το toolchain για τα CMake projects, δηλαδή η διαδικασία του build.

Επιλέξτε το MinGW σύστημα για τα Windows.

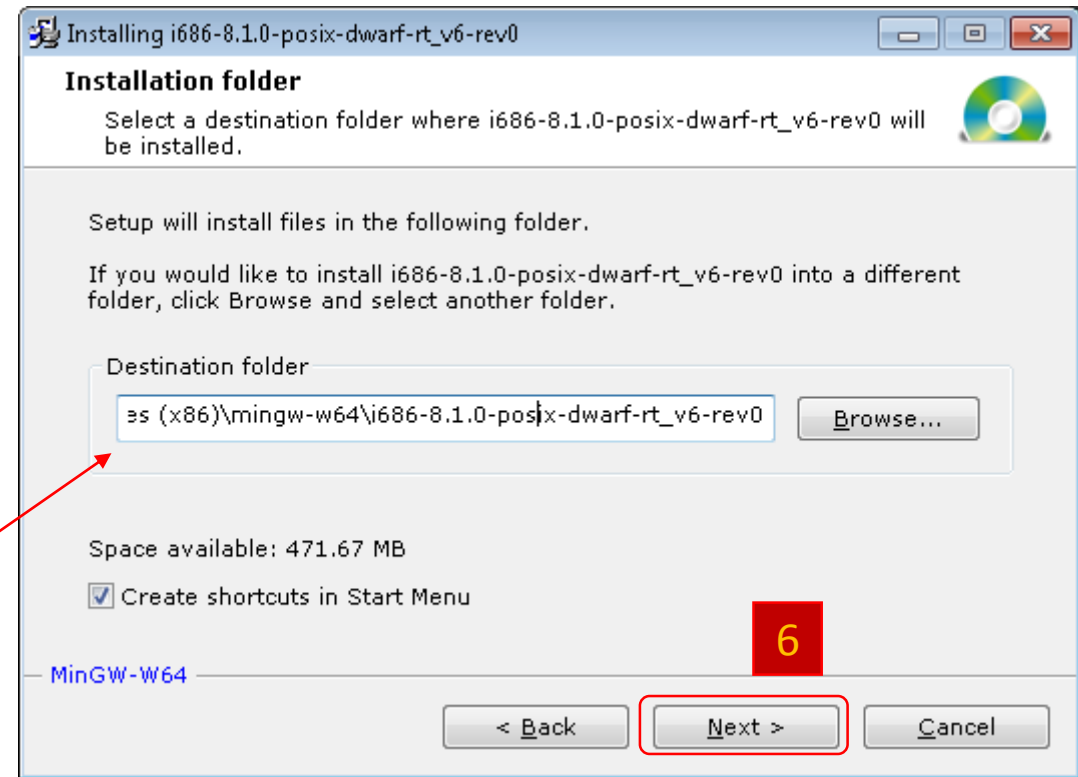
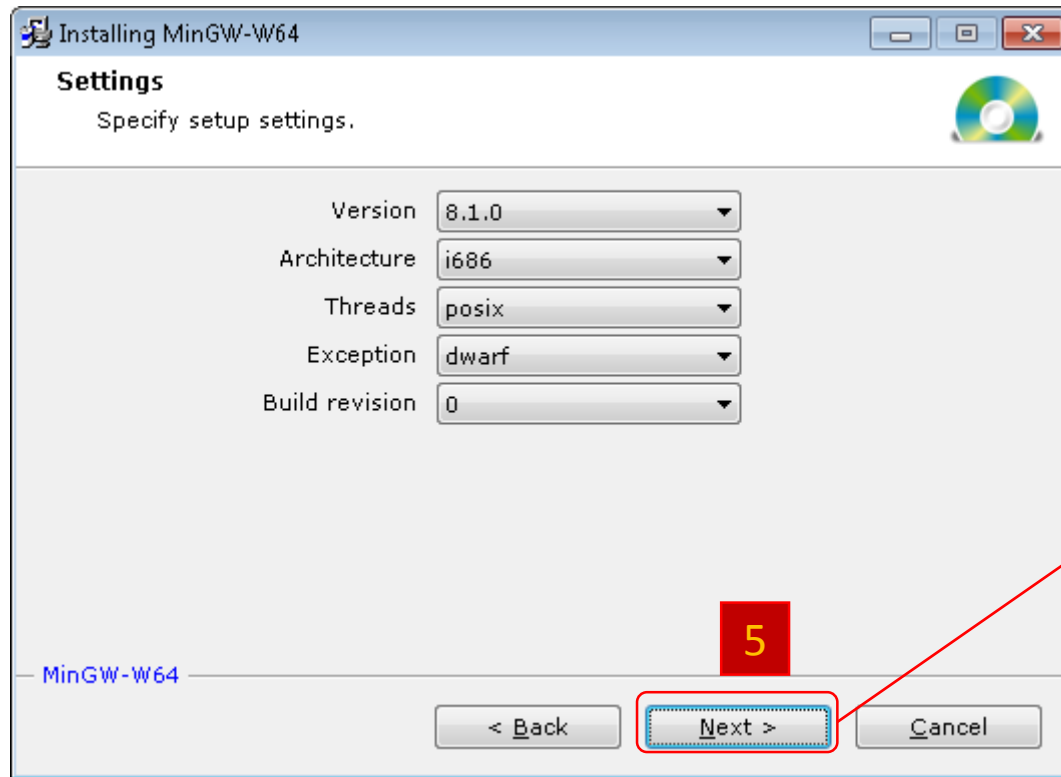
Ακολουθήστε τη διαδικασία εδώ, κανονικά θα σας αρκέσει, αλλά αν χρειαστείτε μπορείτε να δείτε και τις σχετικές σελίδες στο site της JetBrains. [Αυτή](#) για το toolchain και [αυτή](#) για το MinGW.



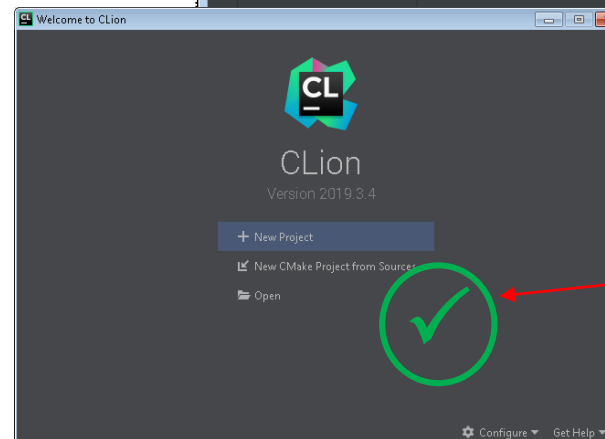
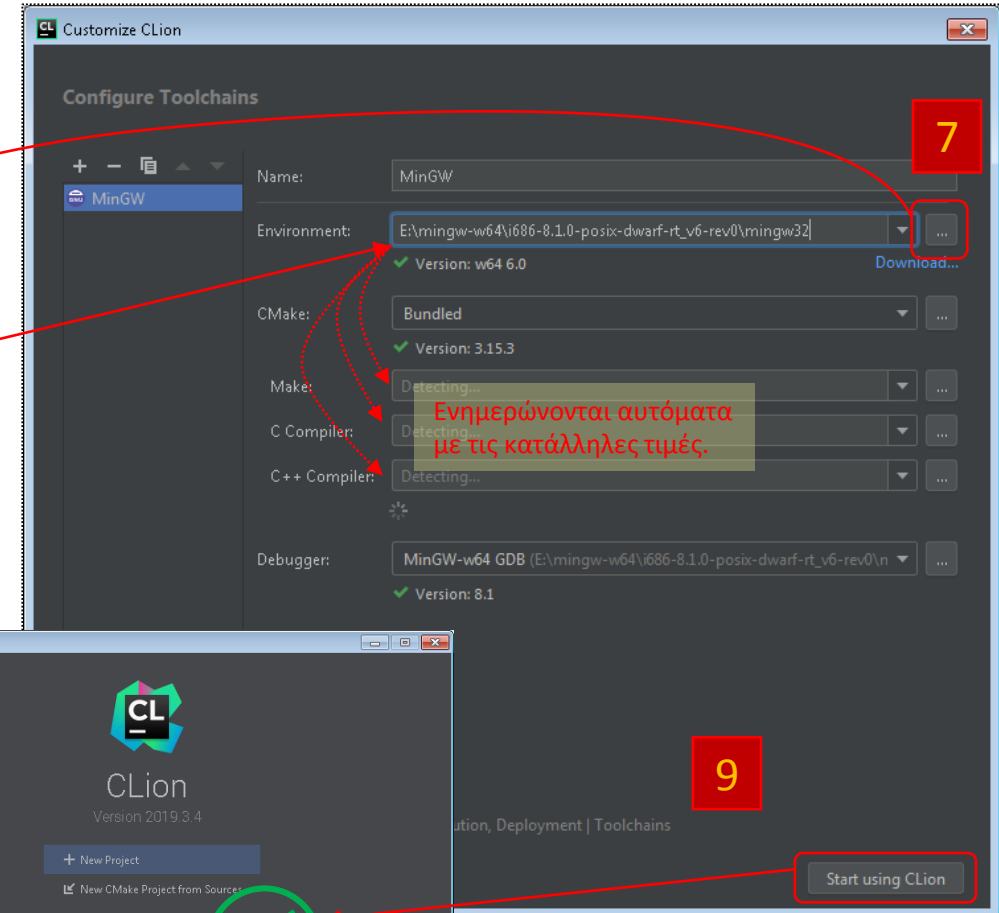
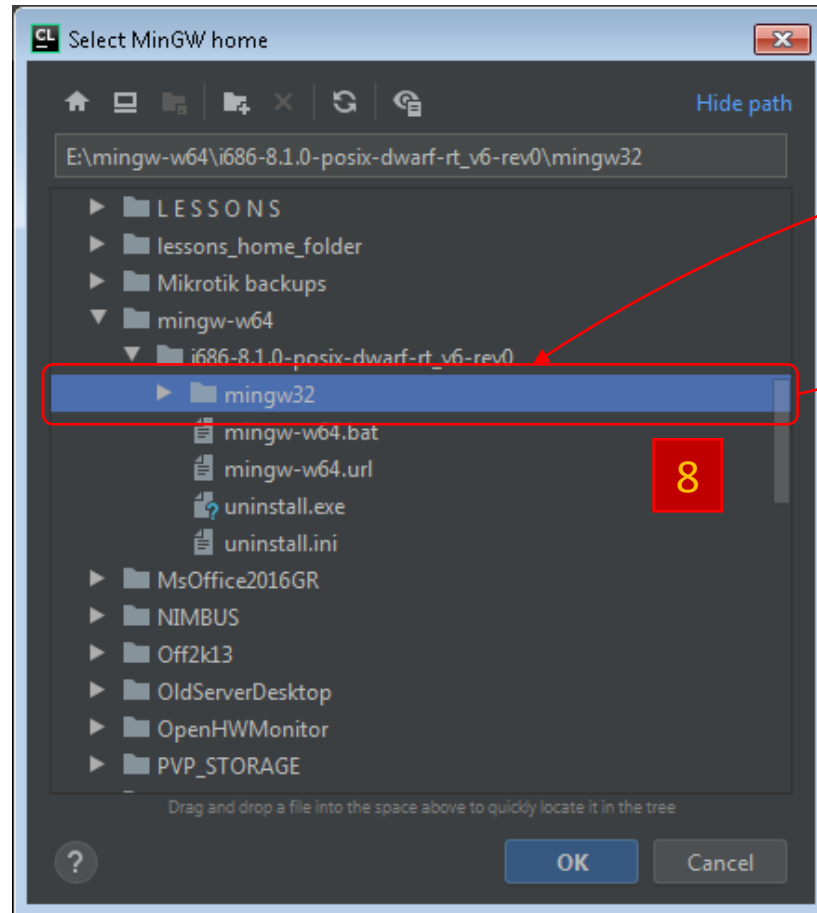
CLion : Ρύθμιση toolchain & MinGW (2/4)



CLion : Ρύθμιση toolchain & MinGW (3/4)



CLion : Ρύθμιση toolchain & MinGW (4/4)



CLion : Οι βασικές περιοχές του IDE

Κεντρικό Menu

Περιλαμβάνει όλες τις επιλογές του IDE. Είναι όλο και πιο χρήσιμο καθώς προοδεύει ο προγραμματιστής

Περιοχή Project

Όλα τα σχετικά και απαραίτητα αρχεία Περιλαμβάνουν και το αρχείο του κώδικα (εδώ main.c)

Καρτέλες Επεξεργαστή

Εμφανίζουν τα αρχεία που είναι ανοιχτά στον επεξεργαστή. Το τρέχον αρχείο ξεχωρίζει.

Γραμμή εκτέλεσης

Περιέχει επιλογή του χτισίματος, της εκτέλεσης, του debugging, κ.α.

Περιοχή Επεξεργαστή

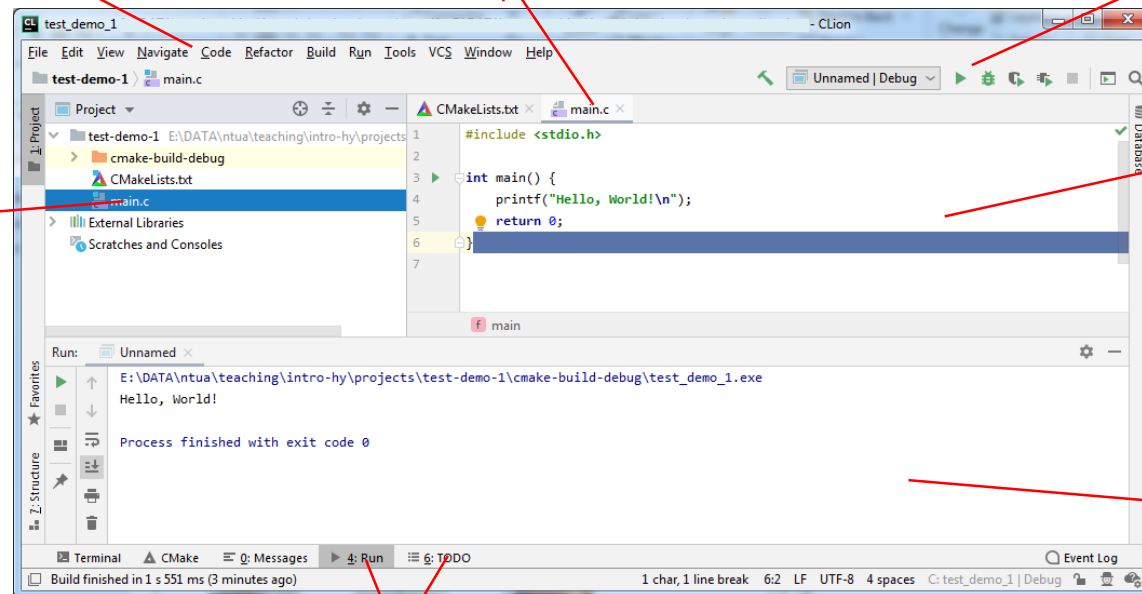
Εμφανίζεται ο κώδικας του προγράμματος. Έχει διάφορα βοηθητικά χαρακτηριστικά όπως είναι ο χρωματισμός των εντολών (syntax highlighting) και άλλα.

Περιοχή Μηνυμάτων & Αποτελεσμάτων

Εμφανίζει τα μηνύματα κατά το Building ή τα μηνύματα κατά την εκτέλεση κ.α.

Καρτέλες Επιλογής

Από αυτές τις καρτέλες επιλέγεται τι εμφανίζει κάθε στιγμή η περιοχή αποτελεσμάτων. Τα δύο βέλη υποδεικνύουν τις δύο πιο συχνά χρησιμοποιούμενες καρτέλες. Η αριστερή εμφανίζει τα μηνύματα κατά τη διάρκεια του build ενώ η δεξιά εμφανίζει τα αποτελέσματα της εκτέλεσης.



CLion : Βασικά στοιχεία του editor

Ο editor (επεξεργαστής) του κώδικα είναι ένας κειμενογράφος απλών κειμένων (όπως το Σημειωματάριο/Notepad των Windows) με αρκετές πρόσθετες δυνατότητες. Για παράδειγμα:

- Syntax highlighting : Χρωματισμός των διαφόρων σημείων του κώδικα ώστε να βοηθά στον εντοπισμό λέξεων κλειδιών, σφαλμάτων, κλπ
- IntelliSense : Αυτόματη συμπλήρωση λέξεων κλειδιών, ονομάτων (identifiers), παραμέτρων συναρτήσεων, κατά την πληκτρολόγηση
- Μετονομασία μεταβλητών, όπου αυτόματα μετονομάζονται όλες οι εμφανίσεις τους
- Προειδοποιήσεις για πιθανά σφάλματα
- Προτάσεις για βελτίωση του κώδικα
- Αυτόματη μορφοποίηση του κώδικα

CLion : Δημιουργία ενός Project

1. Επιλογή «**τι παράγει**»
το project

Επιλέγοντας από το μενού του CLion, *File > New Project* εμφανίζεται το διπλανό πλαίσιο διαλόγου (dialog).

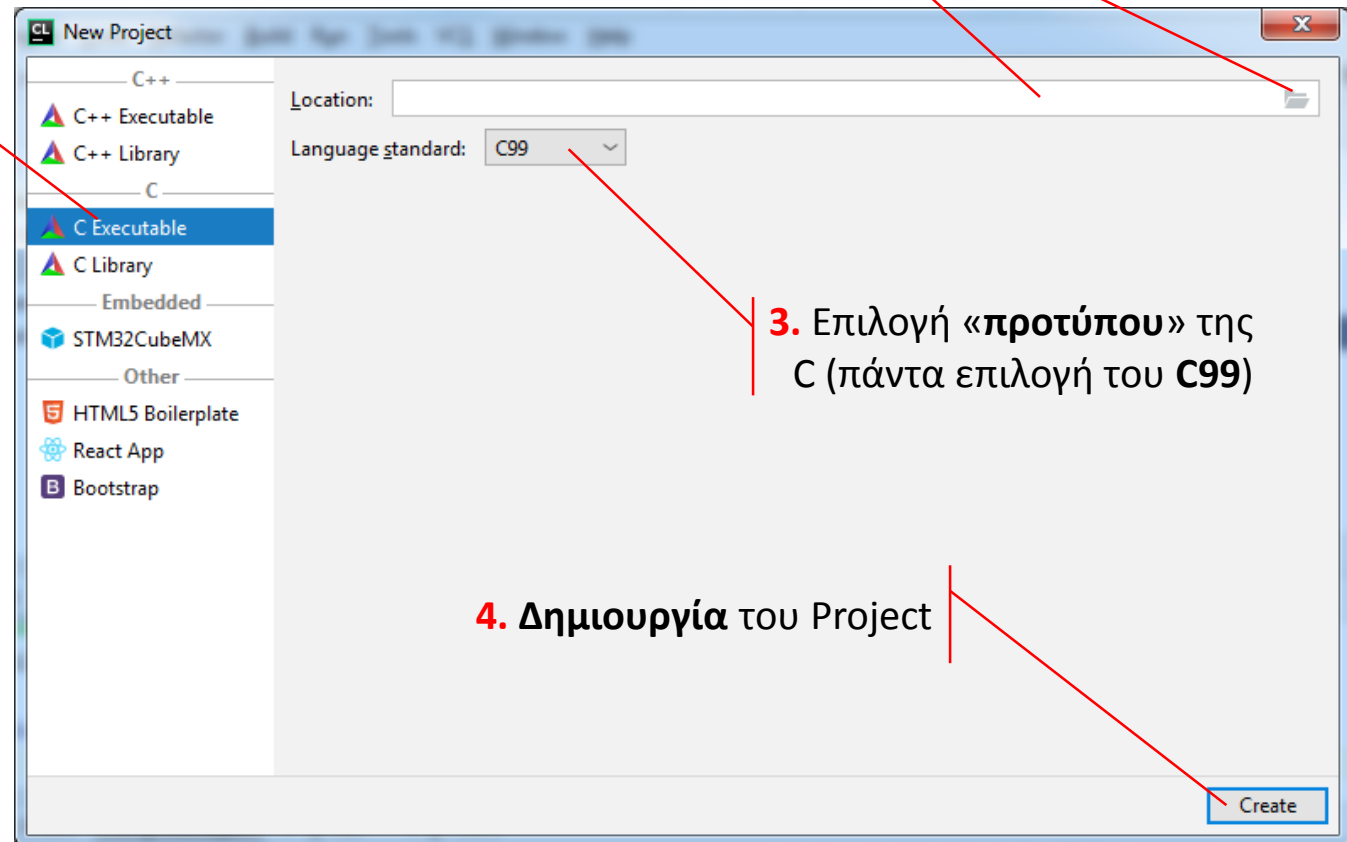
Το κάθε project αποθηκεύεται σε ένα φάκελο στον δίσκο. Εκεί βρίσκονται όλα τα απαραίτητα αρχεία για αυτό το project.

Στο κάθε project αντιστοιχεί τουλάχιστον ένας «στόχος» (target) που είναι το αποτέλεσμα του build. Συνήθως αυτός ο στόχος είναι ένα εκτελέσιμο αρχείο.

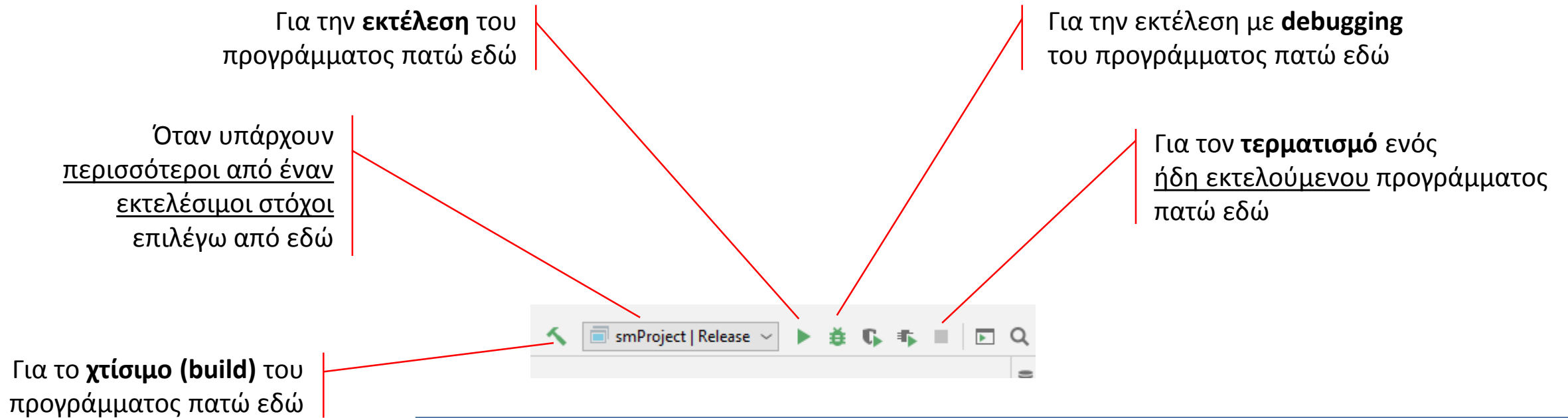
2. Πληκτρολόγηση ή επιλογή του **φακέλου**
αποθήκευσης του Project

3. Επιλογή «**προτύπου**» της
C (πάντα επιλογή του **C99**)

4. Δημιουργία του Project



CLion : Χτίσιμο και Εκτέλεση



Όταν επιλέγει ο χρήστης **εκτέλεση (run)** τότε, εφόσον έχει αλλάξει ο κώδικας από το προηγούμενο build, αυτόματα γίνεται πρώτα **build** το project και κατόπιν ξεκινά η εκτέλεση.

Τα μηνύματα του **build** εμφανίζονται στο κάτω μέρος του IDE στην καρτέλα **Messages**. Ενώ τα μηνύματα της **εκτέλεσης** του προγράμματος εμφανίζονται στην καρτέλα **Run**.

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

<https://qna.c-programming.allos.gr>

- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLion ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

