



ΤΜΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΛΙΤΙΣΜΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ  
& ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

# Αρχές Γλωσσών Προγραμματισμού (C++)

Διάλεξη 2η

Εισαγωγή στον Προγραμματισμό

Γιάννης Θ. Ντόκας

Π.Δ. 407

# Σκοπός της Διάλεξης

2

✓ Να γίνει εισαγωγή στις γλώσσες προγραμματισμού, να κατανοήσουμε τις βασικές έννοιες του προγραμματισμού, και να εξηγήσουμε γιατί επιλέξαμε τη C++.

# Περιεχόμενα της Διάλεξης

3

- ✓ Εισαγωγή στις Γλώσσες Προγραμματισμού
- ✓ Βασικές Έννοιες
- ✓ Ιστορικό Γ.Π.
- ✓ Κατηγορίες Γ.Π

# Εισαγωγή στις Γ.Π.

# Γνωστές Σύγχρονες Ιστορίες

5

Ε  
-  
Σ  
Α  
Γ  
Ω  
Γ  
Η

- **Google** (1996, Larry Page 25, Sergey Brin 24)
  - Σεπτ. 1998: 10,000 search queries/day
  - Δεκ. 2000: 100,000,000 search queries/day
  - Νέες υπηρεσίες: Catalog Search, AdWords, Google News, Local Search, Scholar,...
  - Νέοι πελάτες, παγκοσμιοποίηση,...
- **World Wide Web (WWW)**
  - 1990: Tim Berners-Lee, CERN, Γενεύη
  - 1994: W3C στο MIT μαζί με CERN
  - Υποστήριξη (...) Ευρωπαϊκής Επιτροπής
  - Τώρα: MIT, ERCIM, Keio Univ.
- **Facebook** (2004, Mark Zuckerberg 20)
  - 110 εκ. ενεργοί χρήστες σήμερα
  - 4<sup>ο</sup> πιο δημοφιλές site στον κόσμο
  - Σεπτέμβριος 2007: 50 εκ. χρήστες
  - Ιούνιος 2005: λιγότεροι από 5 εκ. χρήστες
  - Μέσα σε ένα μήνα από την εκκίνησή του, οι μισοί προπτυχιακοί φοιτητές του Harvard ήταν χρήστες.
- **Twitter**, (March 2006, Jack Dorsey)
  - το SMS του Internet
  - πάνω από 500 εκατομμύρια χρήστες
  - πάνω από 340 εκατομμύρια Tweets
  - εφαρμογές για κινητά

# Πίσω από όλα αυτά

6

Ε  
Ι  
Σ  
Α  
Γ  
Ω  
Γ  
Η

- Τεχνολογικές καινοτομίες κυρίως λογισμικού και εφαρμογών – δημοσίευση
- Ελεύθερη διάθεση και χρήση (αρχικά)
- Ικανοποίηση αναγκών χρηστών
- Το WWW είναι και μέσο marketing
- Μέγεθος αγοράς – χωρίς σύνορα
- Προσαρμοστικότητα στο νέο, δυναμικό κοινωνικό και οικονομικό περιβάλλον
- Γλώσσες Προγραμματισμού: [Java](#), [PHP](#), [JavaScript](#), [C++](#), ...
- Νέα εργαλεία παρουσίασης και διάδοσης πληροφορίας: [HTML](#), [CSS](#), [XML](#)
- Νέα περιβάλλοντα προγραμματισμού και ανάπτυξης εφαρμογών: [WWW](#), [Semantic Web](#), [Web Services](#)
- Νέα μέσα γρήγορης μετάδοσης πληροφορίας: [Broadband](#), [Wi-Fi](#), [GSM](#)

# Ερωτήσεις

7

Ε  
Ι  
Σ  
Α  
Γ  
Ω  
Γ  
Η

1. **Γιατί έχουμε** γλώσσες προγραμματισμού; Τι κάνει μια ΓΠ;
  - Τρόπος σκέψης – τρόπος έκφρασης αλγορίθμων
  - Γλώσσες από τη σκοπιά που σκέφτεται ο *χρήστης*
  - *Αφαίρεση* εικονικής μηχανής – τρόπος καθορισμού του *τι θέλουμε*
  - Τρόπος να κάνουμε το hardware να κάνει κάτι *χωρίς* να μπλεχτούμε με τα *bits*
  - Γλώσσες από τη σκοπιά του *κατασκευαστή*

- 2.** ■ Γιατί υπάρχουν τόσες **πολλές** Γλώσσες Προγραμματισμού (ΓΠ);
- *Εξέλιξη* – έχουμε μάθει καλύτερους τρόπους να κάνουμε πράγματα με το πέρασμα του χρόνου
  - *Κοινωνικό-οικονομικοί Παράγοντες*: ιδιωτικά ενδιαφέροντα, εμπορικό πλεονέκτημα
  - *Προσανατολισμός σε ειδικούς σκοπούς*
  - *Προσανατολισμός σε ειδικό hardware*
  - *Διάφορες ιδέες για το τι είναι ευχάριστο να χρησιμοποιείται*



# Ερωτήσεις

9

Ε  
Ι  
Σ  
Α  
Γ  
Ω  
Γ  
Η

- 3.**
- Τι κάνει μια γλώσσα προγραμματισμού **επιτυχημένη**;
    - Ευκολία στη *μάθηση* (BASIC, Pascal)
    - Ευκολία στο να *εκφραστούν* πράγματα, ευκολία στη *χρήση* όταν τη μάθει κανείς (C, Common Lisp, APL, Algol-68, Perl)
    - Ευκολία στην *ανάπτυξη* (BASIC)
    - Δυνατότητα να μεταγλωττιστεί σε *πολύ καλό* (γρήγορο/μικρό) *κώδικα* (Fortran)
    - Να έχει ένα δυνατό *υποστηρικτή* (COBOL, PL/1, Ada, Visual Basic)
    - Μεγάλη διασπορά με *ελάχιστο κόστος* (Pascal, Java)

# Ερωτήσεις

10

Ε  
Ι  
Σ  
Α  
Γ  
Ω  
Γ  
Η

## 4. Γιατί να μάθει κανείς για γλώσσες προγραμματισμού; (1)

■ Βοήθεια για να επιλεγθεί μια γλώσσα

heir contents

- **C vs C++** για προγραμματισμό συστημάτων
- **Fortran vs Ada** για αριθμητικούς υπολογισμούς
- **Common Lisp vs Scheme vs ML** για διαχείριση συμβολικών δεδομένων
- **Java vs C** για δικτυωμένα προγράμματα υπολογιστών

## 4. Γιατί να μάθει κανείς για γλώσσες προγραμματισμού; (2)

- Μερικές γλώσσες είναι παρόμοιες: είναι εύκολο να μάθεις μια γλώσσα αν ξέρεις τη γλώσσα στην οποία βασίστηκε
  - Οι έννοιες έχουν ακόμη μεγαλύτερη ομοιότητα: αν σκεφτεί κανείς τις έννοιες της επανάληψης, αναδρομής, αφαίρεσης, θα βρει πιο εύκολο να αφομοιώσει τη σύνταξη και τις σημασιολογικές λεπτομέρειες μιας νέας γλώσσας από το να προσπαθήσει κάτι από την αρχή. Σε αναλογία με τις ανθρώπινες γλώσσες: μεγάλος έλεγχος της γραμματικής κάνει πιο εύκολη την εκμάθηση μιας νέας γλώσσας (τουλάχιστον ινδοευρωπαϊκή)

4. Γιατί να μάθει κανείς για γλώσσες προγραμματισμού; (3)
  - Βοήθεια για να κάνει κανείς καλύτερη χρήση οποιασδήποτε γλώσσας προγραμματισμού
    - Να κατανοηθούν τα δυσνόητα χαρακτηριστικά  
Π.χ. στη C, βοηθάει να κατανοηθούν ενώσεις, πίνακες και δείκτες, η ξεχωριστή μεταγλώττιση

# Βασικές Έννοιες

# Βασικές Έννοιες

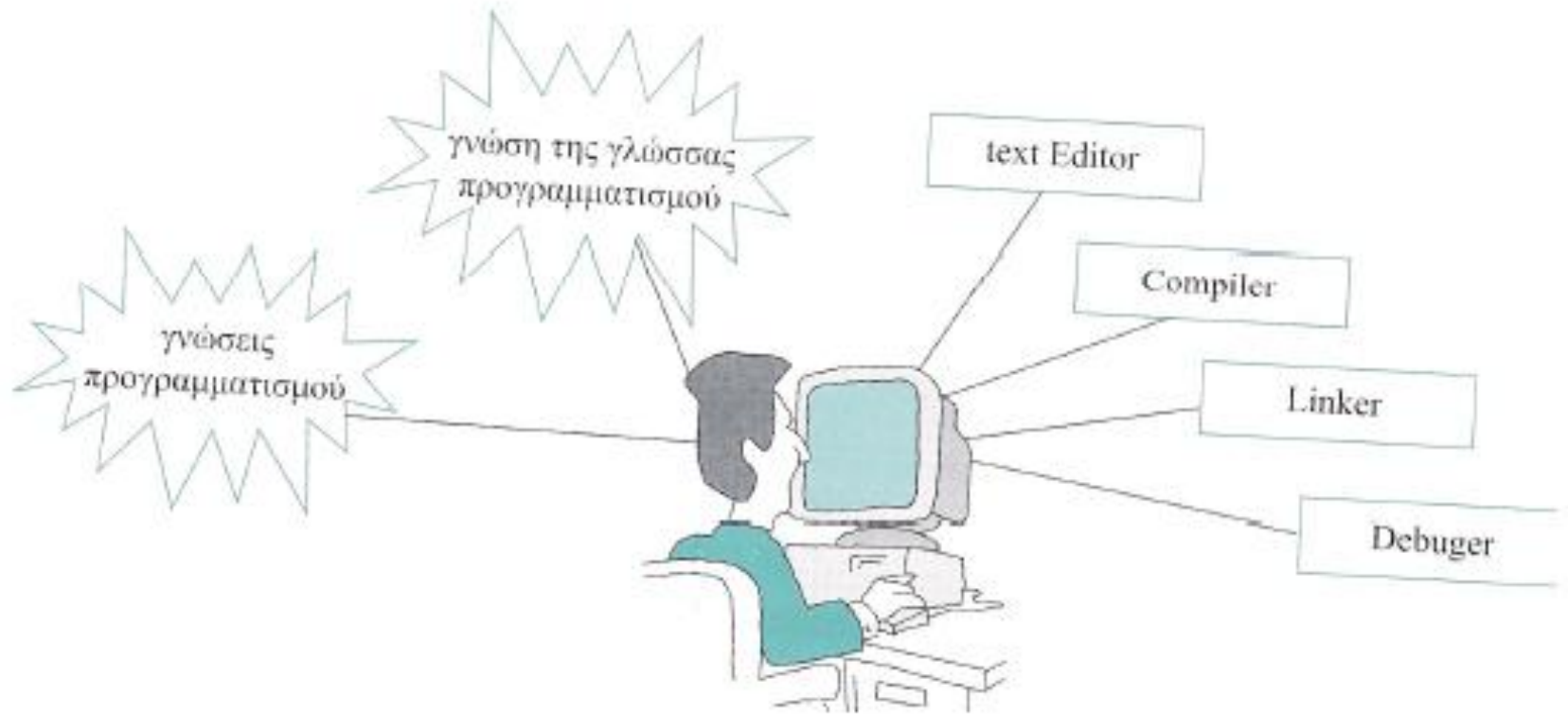
14

- Πρόβλημα (Problem)
  - Το υπολογιστικό πρόβλημα που καλούμαστε να επιλύσουμε, π.χ., ένας αριθμητικός υπολογισμός, η διαχείριση μιας βάσης δεδομένων, ο έλεγχος ενός ρομπότ, μια ιατρική διάγνωση, η πρόγνωση του καιρού.
- Αλγόριθμος (Algorithm)
  - Μια σαφώς καθορισμένη διαδικασία αποτελούμενη από ένα σύνολο εκτελέσιμων βημάτων που θα τερματίσει με ένα αποτέλεσμα μετά από πεπερασμένο πλήθος βημάτων που θα εκτελεσθούν.
- Πρόγραμμα (Program)
  - Μια κωδικοποίηση του αλγόριθμου σε μια γλώσσα προγραμματισμού
- Διεργασία (Process)
  - Το πρόγραμμα που εκτελείται σε έναν υπολογιστή

# Κωδικοποίηση Αλγορίθμου

15

Β  
Α  
Σ  
Ι  
Κ  
Ε  
Σ  
  
Ε  
Ν  
Ν  
Ο  
-  
Ε  
Σ



# Περιεχόμενα Μαθήματος

16

Β  
Α  
Σ  
Ι  
Κ  
Ε  
Σ

Ε  
Ν  
Ν  
Ο  
-  
Γ  
Ε  
Σ

Αλγόριθμος: Διαδικασία επίλυσης προβλήματος.

Γλώσσα Προγραμματισμού: Γλώσσα διατύπωσης αλγορίθμων.

Πρόγραμμα: Αλγόριθμος διατυπωμένος σε κάποια γλώσσα προγραμματισμού.

Γλώσσα  
Προγραμματισμού

- Λεξιλόγιο (σύνολο εντολών)
- Συντακτικό (κανόνες σύνταξης προτάσεων)
- Σημασιολογία (μοντέλο υπολογισμού)



# Ορισμοί

17

Β  
Α  
Σ  
Ι  
Κ  
Ε  
Σ

Ε  
Ν  
Ν  
Ο  
Ι  
Ε  
Σ

- ✓ Γλώσσα προγραμματισμού είναι μια τεχνητή γλώσσα που χρησιμοποιείται για τον έλεγχο μιας μηχανής, συνήθως υπολογιστή
- ✓ Είναι μια συστηματική σημειογραφία με την οποία περιγράφουμε υπολογιστικές διεργασίες
- ✓ Κάθε γλώσσα έχει ένα σύνολο συντακτικών και εννοιολογικών κανόνων που καθορίζουν τη δομή και το νόημα των προτάσεων της
- ✓ Χρησιμοποιείται για την ακριβή διατύπωση αλγορίθμων αλλά και να διευκολύνουν την οργάνωση και διαχείριση πληροφοριών

# Ορισμοί (2)

18

Εντολή για την ανάθεση της τιμής 2 σε μεταβλητή:  
(εξέλιξη)

1. Γλώσσα μηχανής (Intel 8x86, PCs)  
`C7 06 0000 0002` (μετακίνηση του 2 στη δ/νση 0000, 16δικό)
2. Assembly (εξαρτώμενη από μηχανή)  
`MOV X, 2`
3. Γλώσσα Προγραμματισμού Υψηλού Επιπέδου  
`X = 2`

# Ιστορικό Γ.Π.

## ■ Δεκαετίες 1930 & 1940

Πολλές ΓΠ που δεν είχαν στόχο την υλοποίηση:

- **Plankalkül** (Zuse Konrad)

$$\begin{array}{l|l} & A + 1 \rightarrow A \\ v & 4 \quad \quad 5 \\ s & 1.n \quad \quad 1.n \end{array} \quad \leftrightarrow \quad A(5) := A(4) + 1$$

- **Turing Machine** (Alan Turing)
- **Lambda Calculus** (Alonzo Church)
- **Mark I** (Howard Aiken)
- **Διαγράμματα Ροής** (Von Neumann)

# Ιστορικά

21

- Μέσα δεκαετίας του 1950
  - **FORTAN** (FORmula TRANslation)
    - IBM (1954) για τον IBM 704, John Backus (Υλοποίηση 1957)
    - Χωρίς συναρτήσεις του χρήστη
    - Μεταβλητές 2 char
    - If <συνθήκη> 8, 10 (δηλ. If <συνθήκη> true goto 8, else goto 10)
- Αρχές δεκαετίας του 1960
  - **ALGOL 60**
    - Αλγοριθμική γλώσσα περιγραφής υπολογιστικών διεργασιών
    - Επιτακτική (imperative): Αλγόριθμος ως ακολουθία αλλαγών μνήμης
    - Βασικές μονάδες υπολογισμού: blocks, procedures
    - Έννοιες «Τύπου» και «Ελέγχου Τύπων»
    - Κανόνες λεξικής εμβέλειας

# Ιστορικά

22

- **COBOL** (COmmon Business Oriented Language)  
1960, Υπ. Άμυνας ΗΠΑ
  - Περιγραφή δεδομένων ανεξάρτητη από μηχανή (Συστήματα Διαχείρισης Βάσεων Δεδομένων)
  - Γενική if then else
  - Σχόλια
- Δεκαετία 1960 και στη συνέχεια... (βάση η ALGOL)
  - **BASIC** (Beginners All purpose Symbolic Instruction Code)  
1964, Dartmouth College, USA (Interpreter το 1975 η Microsoft)
  - **Pascal**  
1971, Niklaus Wirth
  - **C**  
1972, Bell Labs, Dennis Ritchie (για UNIX system programming)

# Ιστορικά

23

- C++  
1979, Bell Labs, Bjarne Stroustrup
- Java  
1991, Sun
- HTML (HyperText Markup Language) – όχι ακριβώς ΓΠ...  
1991, CERN, Tim Berners-Lee
- PHP (Hypertext Preprocessor)  
1994, Rasmus Lerdorf
- JavaScript  
1995, Netscape

Εντωμεταξύ:

Ada, APL, LISP, SNOBOL, PL/1, SIMULA, ALGOL-W, Scheme, ML, Prolog, Smalltalk, VisiCalc, RPG, Perl, ...

# Ομάδες Γ.Π.



# Ομάδες Γλωσσών

25

Ο  
Μ  
Α  
Δ  
Ε  
Σ

Γ.

Π.

- ✓ Ανάλογα με το δομή του προγράμματος
  - ❑ Procedural (Διαδικαστικές), π.χ. C, Pascal
  - ❑ Object Oriented (Αντικειμενοστραφής), π.χ. C++, Java
  - ❑ Functional (Συναρτησιακές), Lisp, Haskell
  
- ✓ Ανάλογα με το στόχο που έχει η γλώσσα
  - ❑ Γενικής Χρήσης π.χ. Pascal
  - ❑ Προγραμματισμού συστημάτων, π.χ. C++
  - ❑ Scripting, π.χ. Perl
  - ❑ Ειδικών εφαρμογών, π.χ. Matlab
  - ❑ Παράλληλες ή κατανεμημένες, π.χ. Java

# Ομάδες Γλωσσών

26

Ο  
Μ  
Α  
Δ  
Ε  
Σ

Γ.  
Π.

- ✓ Ανάλογα με την περιγραφή του ζητούμενου αποτελέσματος
  - Imperative (Προστακτικά), π.χ. C, C++, Java
  - Declarative (Δηλωτικά ?), π.χ. Perl

# Ομάδες Γλωσσών

27

Ο  
Μ  
Α  
Δ  
Ε  
Σ

Γ.  
Π.

## ■ Επιτακτικές (Imperative) Γλώσσες Προγραμματισμού

Von Neumann	FORTRAN, Pascal, BASIC, C
Αντικειμενοστραφείς	C++, Java, Smalltalk
Γλώσσες Περιγραφής Σεναρίων (Scripting Languages)	Javascript, PHP, Perl

## ■ Γλώσσες Εφαρμοστικού (Applicative) Προγραμματισμού

Συναρτησιακές (Functional)	Scheme, ML, LISP
Λογικές (Logic)	Prolog, VisiCalc, RPG

# Επιτακτικές Γλώσσες

28

Ο  
Μ  
Α  
Δ  
Ε  
Σ

Γ.  
Π.

- Ο υπολογισμός επιτυγχάνεται με αλλαγή κατάστασης μεταβλητών (θέσεις μνήμης).
- Οι πιο δημοφιλείς ΓΠ.
- Δυνατότητες:
  - Ενοποίηση ομάδας εντολών (begin-end, { })
  - Περιγραφή ομάδων δεδομένων (arrays, records)
  - Περιγραφή ροής ελέγχου (while, for, case)
  - Μηχανισμοί αφαίρεσης (abstraction) δεδομένων (function, procedure)
- Δεν εκμεταλλεύονται πλήρως τις δυνατότητες των Η/Υ.  
Επεξεργασία μιας λέξης τη φορά

# Συναρτησιακές

29

- Βασίζονται σε μαθηματικές συναρτήσεις
- Δεδομένα και προγράμματα έχουν την ίδια μορφή
- Δεν χρησιμοποιούνται «μεταβλητές» και εντολές ανάθεσης
- Επανάληψη με χρήση αναδρομής

**LISP:**

```
(print(process_data(get_data(...))))
```



begin
GetData(...);
ProcessData(...);
OutputData(...);
end.

# Λογικές

30

- Προγράμματα σε μορφή συμβολικής λογικής
- Δηλώνονται οι προδιαγραφές των επιθυμητών αποτελεσμάτων
- Ουσιαστικά δημιουργούνται Βάσεις Δεδομένων
- Προγράμματα: facts, rules, queries

## Prolog:

**FACT:** mother (Helen, Jim)

**RULE:** grandparent (X, Z) : -parent (X, Y), parent (Y, Z)

**QUERY:** father (Bob, Joe)

# Κυριότερες γλώσσες

31

Ο  
Μ  
Α  
Δ  
Ε  
Σ

Γ.  
Π.

Category	Ratings Feb 2011	Delta Feb 2010
Object-Oriented Languages	57.7%	+4.2%
Procedural Languages	37.0%	-5.0%
Functional Languages	4.1%	+1.0%
Logical Languages	1.2%	-0.2%

*By TIOBE  
Programming  
Community  
Index 2011*

Programming Language	Position Feb 2011	Position Feb 2006	Position Feb 1996	Position Feb 1986
<a href="#">Java</a>	1	1	-	-
<a href="#">C</a>	2	2	1	1
<a href="#">C++</a>	3	3	3	8
<a href="#">Python</a>	4	8	22	-
<a href="#">PHP</a>	5	5	-	-
<a href="#">C#</a>	6	7	-	-
<a href="#">(Visual) Basic</a>	7	4	2	5
<a href="#">Objective-C</a>	8	46	-	-
<a href="#">JavaScript</a>	9	10	-	-
<a href="#">Perl</a>	10	6	7	-
<a href="#">Lisp</a>	13	14	5	3
<a href="#">Ada</a>	20	17	10	2

# Γιατί C++ ?



# C++ και C

33

Γ  
Λ  
Ω  
Σ  
Σ  
Ε  
Σ  
Π  
Ρ  
Ο  
Γ  
Ρ  
Α  
Μ  
Μ  
Α  
Τ.

- ✓ Η C++ είναι εξέλιξη της C και για την ακρίβεια είναι υπερσύνολο της C
- ✓ Κάθε σωστή πρόταση στη C είναι πρόταση στη C++
- ✓ Η κυριότερη διαφορά μεταξύ C και C++ είναι η εισαγωγή των κλάσεων και αντικειμένων στη C++
- ✓ Βελτιωμένη προσέγγιση εισόδου και εξόδου δεδομένων
- ✓ Δημιουργία ευέλικτου κώδικα με δυνατότητα επαναχρησιμοποίησης
- ✓ Τα πλεονεκτήματα της C++ σε σχέση με την C είναι αντίστοιχα με τα πλεονεκτήματα του OOP σε σχέση με με τον PP και συγκεκριμένα:

34

# Back Up

# Πλατφόρμα DevC++

35

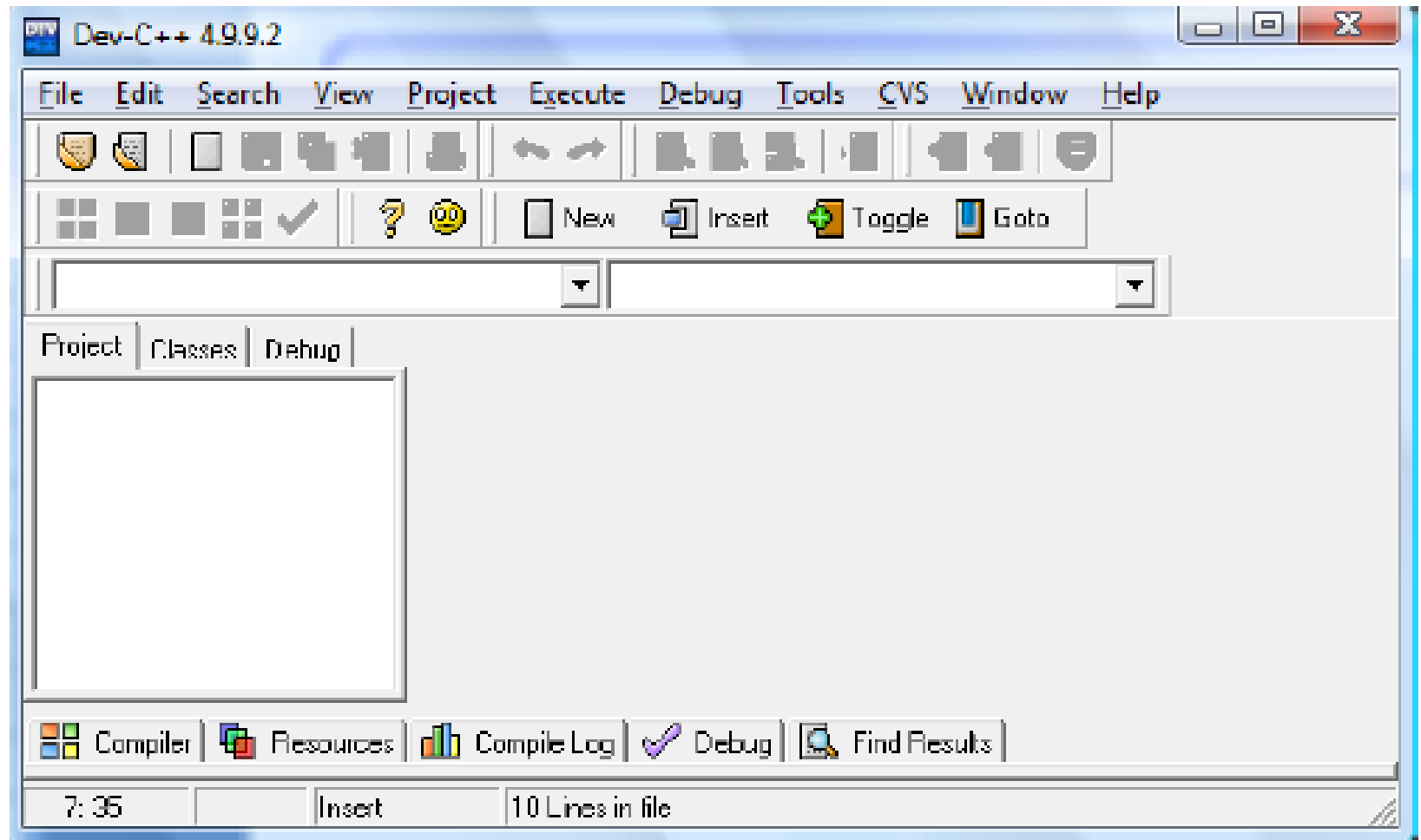
B  
A  
C  
K  
  
U  
P

- Ανοικτό λογισμικό (GNU General Public License)
- **Integrated Development Environment** για C/C++ προγράμματα
- Χρησιμοποιεί την MinGW υλοποίηση του gcc
- Η τελευταία έκδοση μπορεί να βρεθεί στην παρακάτω διεύθυνση:  
<http://www.bloodshed.net/dev/devcpp.html>
  - Dev-C++ 5.0 beta 9.2 (4.9.9.2) (9.0 MB) with Mingw/GCC 3.4.2
  - Κατεβάζετε ένα setup αρχείο της μορφής devcpp-4.9.9.2\_setup.exe

# Το περιβάλλον της DevC++

36

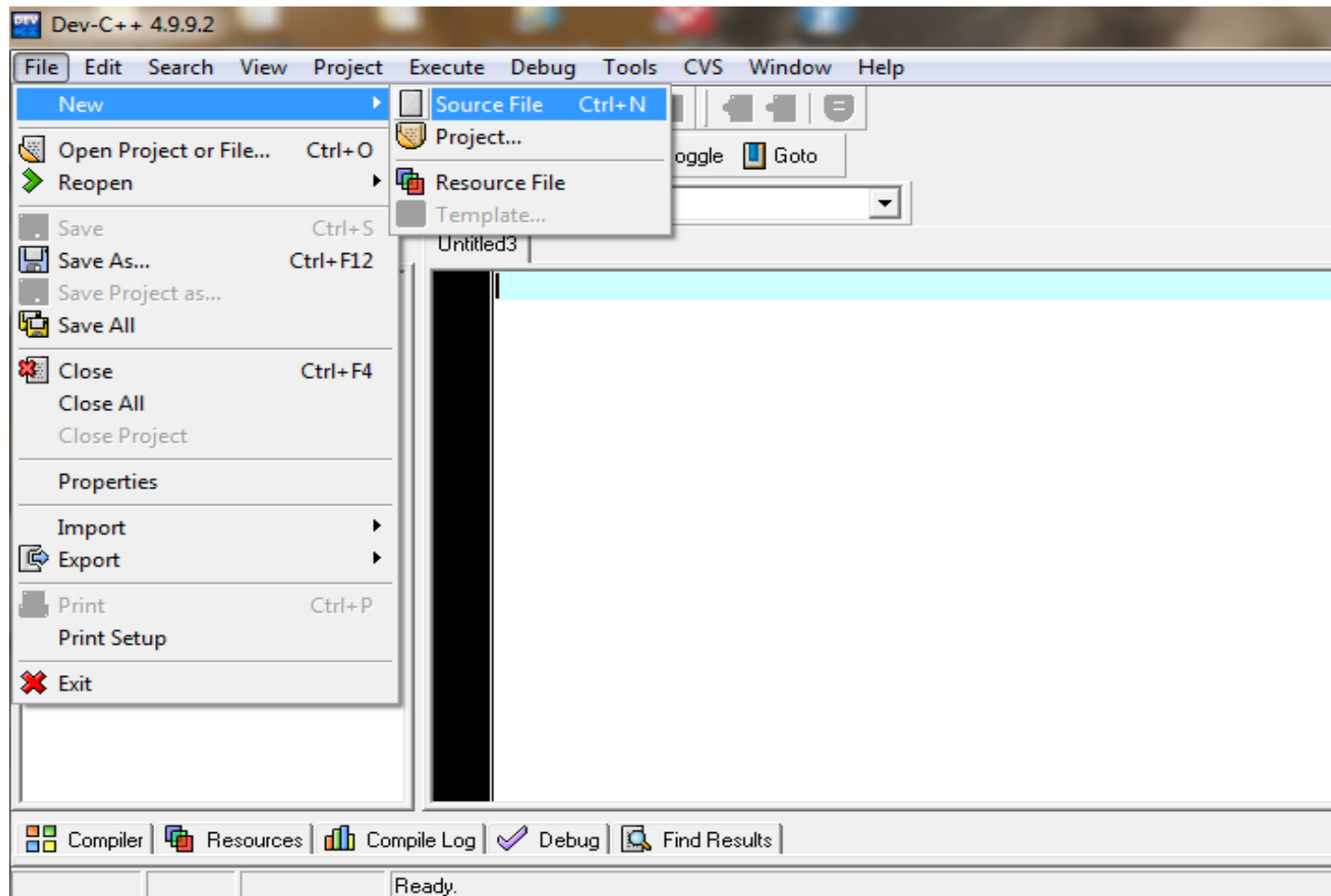
B  
A  
C  
K  
  
U  
P



# Δημιουργία Source File

37

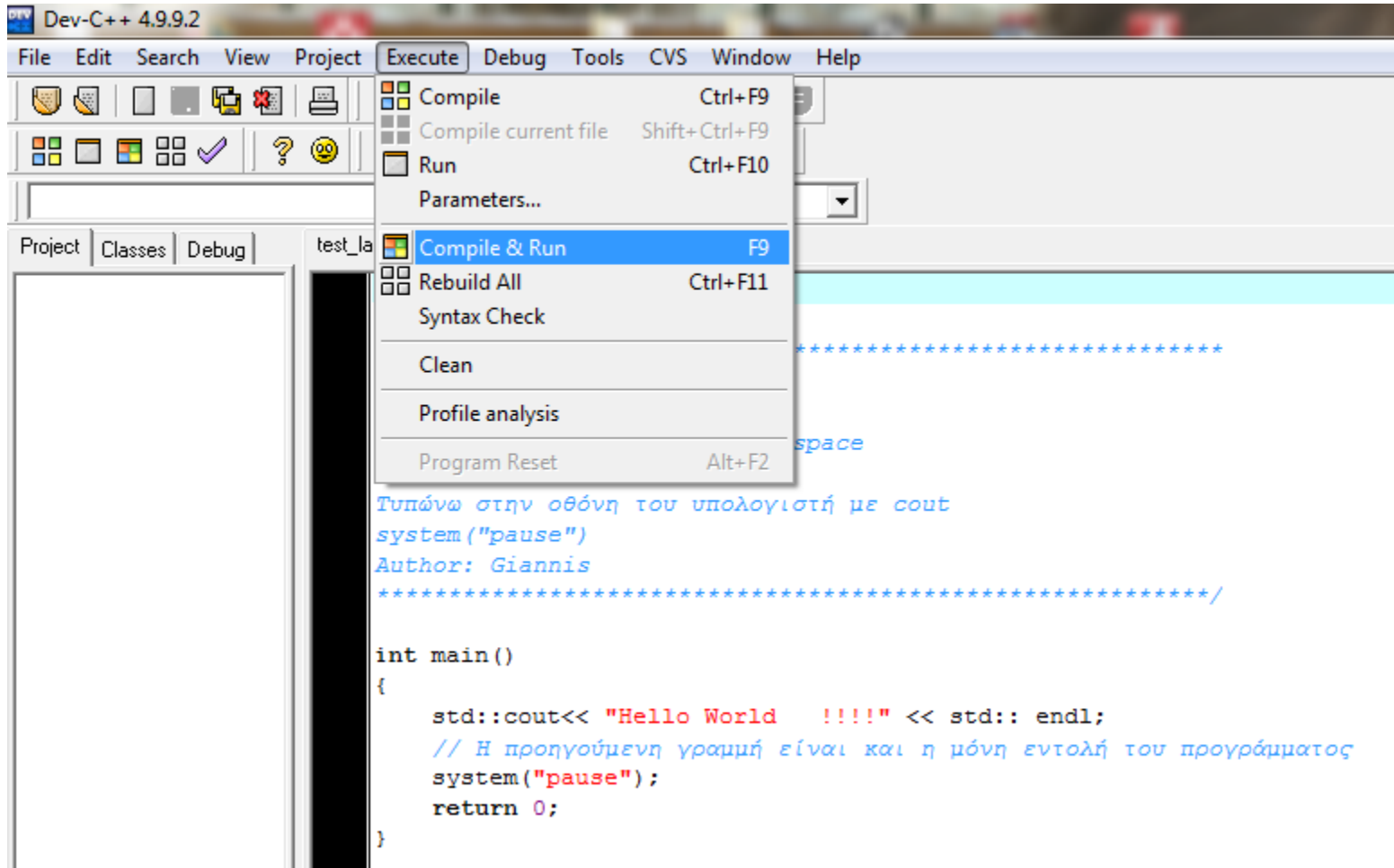
B  
A  
C  
K  
  
U  
P



# Compile & Run

38

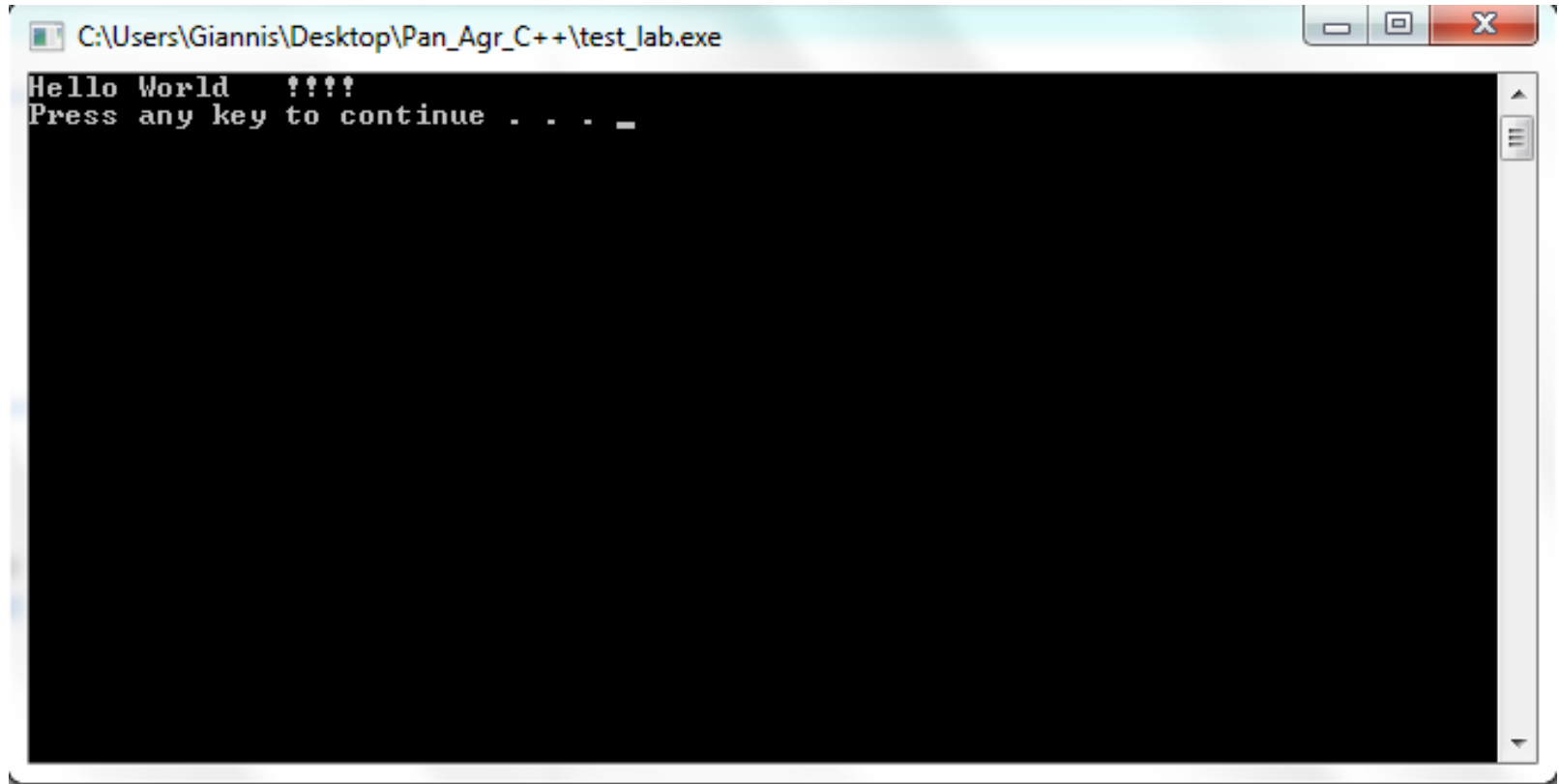
Δ  
Ι  
Δ  
Α  
Κ  
Τ  
Ε  
Α  
Υ  
Λ  
Η



# Αποτέλεσμα Προγράμματος

39

Δ  
Ι  
Δ  
Α  
Κ  
Τ  
Ε  
Α  
Υ  
Λ  
Η

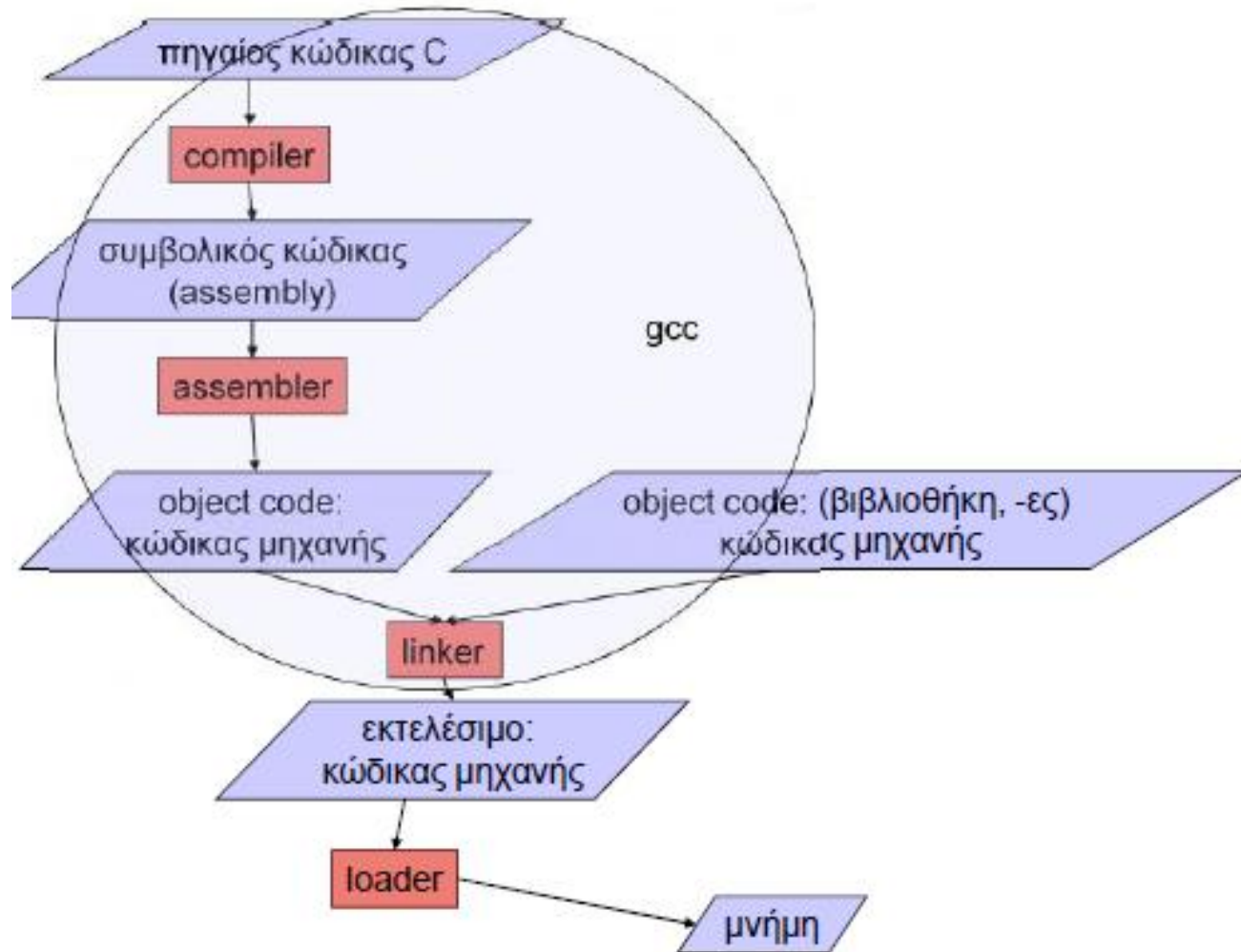


```
C:\Users\Giannis\Desktop\Pan_Agr_C++\test_lab.exe
Hello World  !!!!!
Press any key to continue . . . _
```

# Βήματα Μετάφρασης Κώδικα

40

B  
A  
C  
K  
  
U  
P

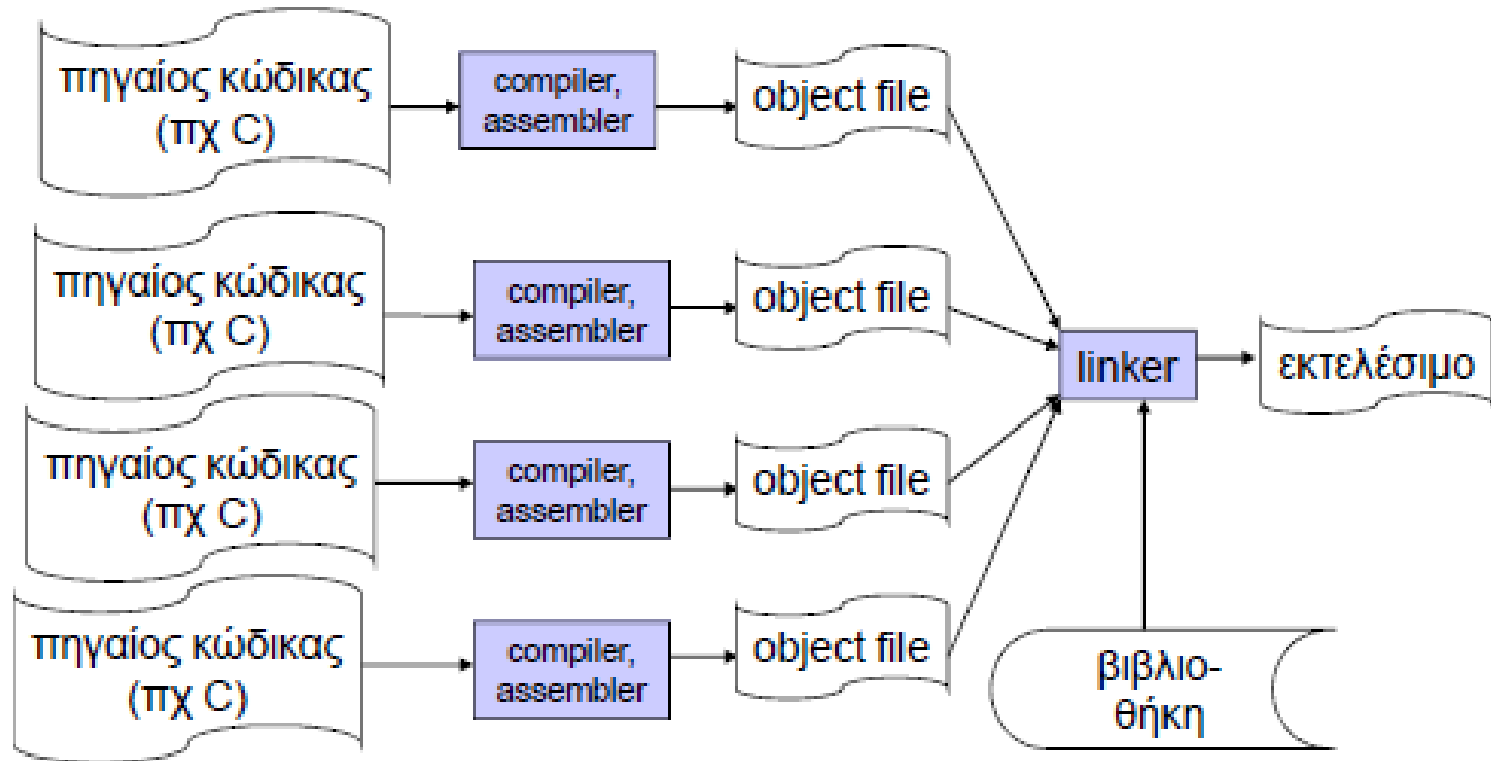




# Δημιουργία εκτελέσιμου

41

B  
A  
C  
K  
  
U  
P



# Κριτήρια καλού σχεδιασμού

42

B  
A  
C  
K  
  
U  
P

- Καλή συντακτική και σημασιολογική περιγραφή
- Αξιοπιστία
- Γρήγορη μετάφραση
- Αποδοτικός αντικειμενικός κώδικας
- Ορθογωνιότητα
- Ανεξαρτησία από μηχανή
- Τεκμηρίωση
- Συνέπεια

# Για την επιτυχία του προγράμματος

43

B  
A  
C  
K  
  
U  
P

- Απλότητα
- Σαφήνεια συντακτικού
- Υποστήριξη αφαιρετικότητας
- Ευκολία πιστοποίησης ορθότητας προγραμμάτων
- Περιβάλλον ανάπτυξης εφαρμογών
- Κόστος χρήσης
  - εκτέλεση
  - μετάφραση
  - δημιουργία
  - συντήρηση