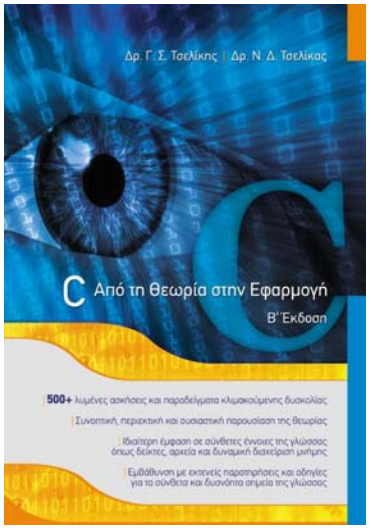
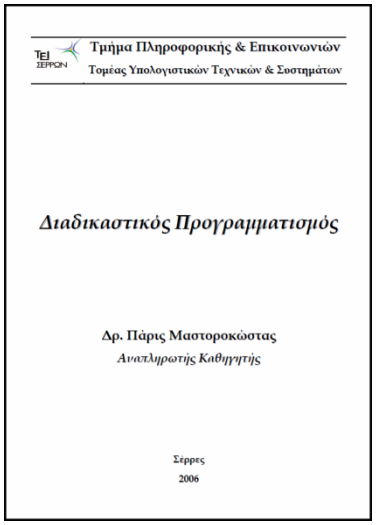
Διδάσκων: Δημήτριος Βαρσάμης

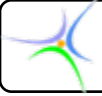
dvarsam@teicm.gr

http://teachers.teicm.gr/dvarsam/

Διανεμόμενα συγγράμματα:







Εισαγωγή – Βασικά στοιχεία προγράμματος

Τι είναι ο υπολογιστής;

- **Οι κανόνες αποτελούν την καρδιά της επιστήμης, της κοινωνίας και της ισχύος σε οιαδήποτε μορφή.**
- **Ο υπολογιστής είναι *μία ταχεία, ακούραστη μηχανή που ακολουθεί κανόνες*.**

Τι είναι ο προγραμματισμός

- **Πρόγραμμα: Ακολουθία εντολών, με τις οποίες ο υπολογιστής εκτελεί μία συγκεκριμένη εργασία και επιλύει ένα δοθέν πρόβλημα.**
- **Προγραμματισμός: Κατάστρωση και συγγραφή προγραμμάτων.**

Τι θα μάθετε

1) Να λύνετε προβλήματα που περιγράφονται από κανόνες: γράφοντας σύνολα κανόνων (συναρτήσεις), οι οποίοι εφαρμόζονται σε σύνολα δεδομένων (δομές), χρησιμοποιώντας ταυτόχρονα και έτοιμα προγράμματα (βιβλιοθήκες)

2) Θεμελιώδεις έννοιες, κοινές σε όλες τις γλώσσες προγραμματισμού

3) Εξοικείωση με ένα αποδοτικό Εργαλείο Προγραμματισμού:
   Το περιβάλλον προγραμματισμού της C++

Προγραμματισμός: Κοινή λογική και καλλιτεχνία

1) Ανάλυση: (εύρεση του πυρήνα του προβλήματος)

- **Καθόρισε τις συγκεκριμένες εισόδους και εξόδους**
- **Καθόρισε τη σχέση εισόδου - εξόδου**
- **Τεμάχισε το πρόβλημα σε υποπροβλήματα**

2) Υλοποίηση: Γράψε το σύνολο κανόνων για κάθε υποπρόβλημα

3) Debug: Έλεγξε κάθε τμήμα ξεχωριστά. Στη συνέχεια συνέδεσέ τα και έλεγξέ τα έως ότου το συνολικό πρόγραμμα λειτουργήσει σωστά.

Ο πυρήνας ενός προβλήματος

Παράδειγμα: **Να γραφεί πρόγραμμα που υπολογίζει το εμβαδό ενός σπιτιού, με δεδομένες τις διαστάσεις κάθε δωματίου.**

- **Είσοδοι:**
  - **Ο αριθμός των δωματίων (ακέραιος αριθμός)**
  - **Οι διαστάσεις κάθε δωματίου (πραγματικοί αριθμοί)**
- **Έξοδοι:**
  - **Το εμβαδό του σπιτιού (πραγματικοί αριθμοί)**

Ανάλυση: Βρείτε τον πυρήνα…

- **Καταστείτε σίγουροι ότι καταλάβατε πραγματικά το πρόβλημα!**
  - **Απλοποιείστε το, διευκρινίστε τα σκοτεινά σημεία, σκεφθείτε. Επαναλάβατε.**
  - **Τεμαχίσατέ το σε υποπροβλήματα.**
  - **Οι πρώτες σκέψεις δεν είναι ΠΟΤΕ οι καλύτερες.**
- **Παράδειγμα σπιτιού:**
  - **Υποπρόβλημα 1: υπολογισμός του εμβαδού κάθε δωματίου**
  - **Υποπρόβλημα 2: πρόσθεση των εμβαδών των δωματίων**

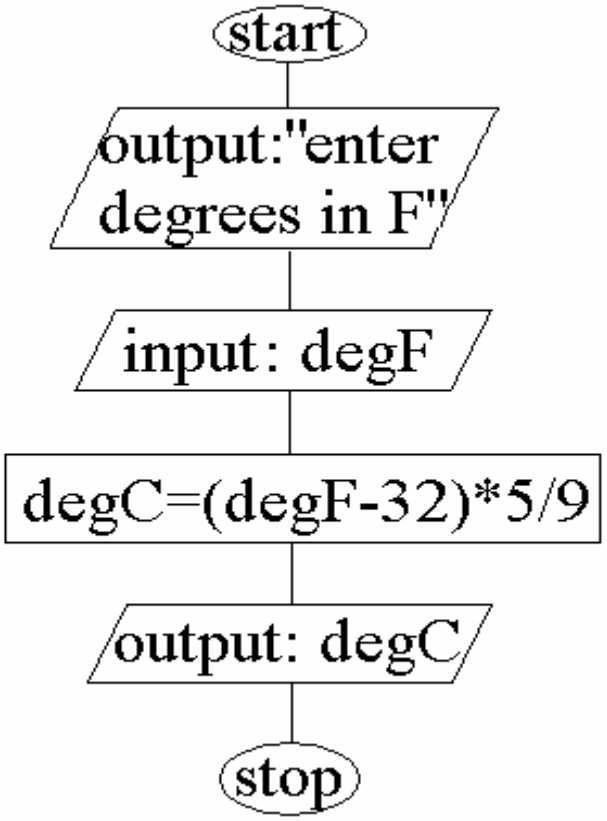
Ανάλυση: Βρείτε τον πυρήνα…

- **Εργαλεία που βοηθούν στην περιγραφή του προβλήματος**
  - **Φυσική γλώσσα: "κάνε αυτό, στη συνέχεια εκείνο, εκτός εάν, κ.λ.π."**
  - **Διάγραμμα ροής**
  - **Ψευδοκώδικας**
- ***Παράδειγμα*: Να μετατραπούν οι βαθμοί Farenheit σε βαθμούς Κελσίου**

Εργαλείο ανάλυσης: η φυσική γλώσσα

- Ζήτησε από το χρήστη τη θερμοκρασία σε βαθμούς *F*.
- Διάβασε την τιμή που δίνει ο χρήστης.
- Αποθήκευσε την τιμή σε θέση αποθήκευσης που καλείται **degF**.
- Υπολόγισε τους βαθμούς *C* με χρήση μαθηματικής σχέσης.
- Αποθήκευσε το αποτέλεσμα σε θέση αποθήκευσης που καλείται **degC**.
- Τύπωσε το περιεχόμενο της **degC**.

Εργαλείο ανάλυσης: το διάγραμμα ροής



Προτιμητέο εργαλείο για σύνθετα προβλήματα με περιορισμούς

Εργαλείο ανάλυσης: ο ψευδοκώδικας

```
print “enter degrees in Farenheit”
read degF
degC = (degF - 32) * 5 / 9
print degC
```

Υλοποίηση

Μετατροπή του συνόλου κανόνων σε συντακτικό της γλώσσας C.

<u>Ο τρόπος μετατροπής εξάγεται διερευνώντας:</u>

Μεταβλητές, τύπους δεδομένων, εκφράσεις, υποθετικές προτάσεις και προτάσεις ελέγχου ροής, συναρτήσεις κ.ά.

Χαρακτηριστικά της γλώσσας C

1) Μπορεί να χρησιμοποιηθεί και ως γλώσσα προγραμματισμού χαμηλού επιπέδου, επιτρέποντας άμεση πρόσβαση στους πόρους του υπολογιστή.

2) Είναι σχετικά μικρή και εύκολη στην εκμάθηση.

3) Υποστηρίζει δομημένο προγραμματισμό.

4) Είναι αποτελεσματική, παράγοντας συμπαγή και γρήγορα στην εκτέλεση προγράμματα.

5) Αποτελεί μαζί με τη C++ τις ευρύτερα χρησιμοποιούμενες γλώσσες σε ερευνητικά και αναπτυξιακά προγράμματα, γεγονός που έχει δημιουργήσει μία πολλή μεγάλη εγκατεστημένη βάση εφαρμογών που αναπτύχθηκαν με αυτές τις γλώσσες και πρέπει να συντηρούνται και να εξελίσσονται.

Sep 2014	Sep 2013	Change	Programming Language	Ratings	Change
1	1		C	16.721%	-0.25%
2	2		Java	14.140%	-2.01%
3	4	↑	Objective-C	9.935%	+1.37%
4	3	↓	C++	4.674%	-3.99%
5	6	↑	C#	4.352%	-1.21%
6	7	↑	Basic	3.547%	-1.29%
7	5	↓	PHP	3.121%	-3.31%
8	8		Python	2.782%	-0.39%
9	9		JavaScript	2.448%	+0.43%
10	10		Transact-SQL	1.675%	-0.32%
11	11		Visual Basic .NET	1.532%	-0.31%
12	12		Perl	1.369%	-0.32%
13	13		Ruby	1.281%	-0.10%
14	-	⇈	Visual Basic	1.272%	+1.27%
15	14	↓	Delphi/Object Pascal	1.157%	+0.26%
16	26	⇈	F#	0.990%	+0.49%
17	15	↓	Pascal	0.893%	+0.01%
18	-	⇈	Swift	0.852%	+0.85%
19	19		MATLAB	0.818%	+0.18%
20	17	↓	PL/SQL	0.809%	+0.13%

www.tiobe.com

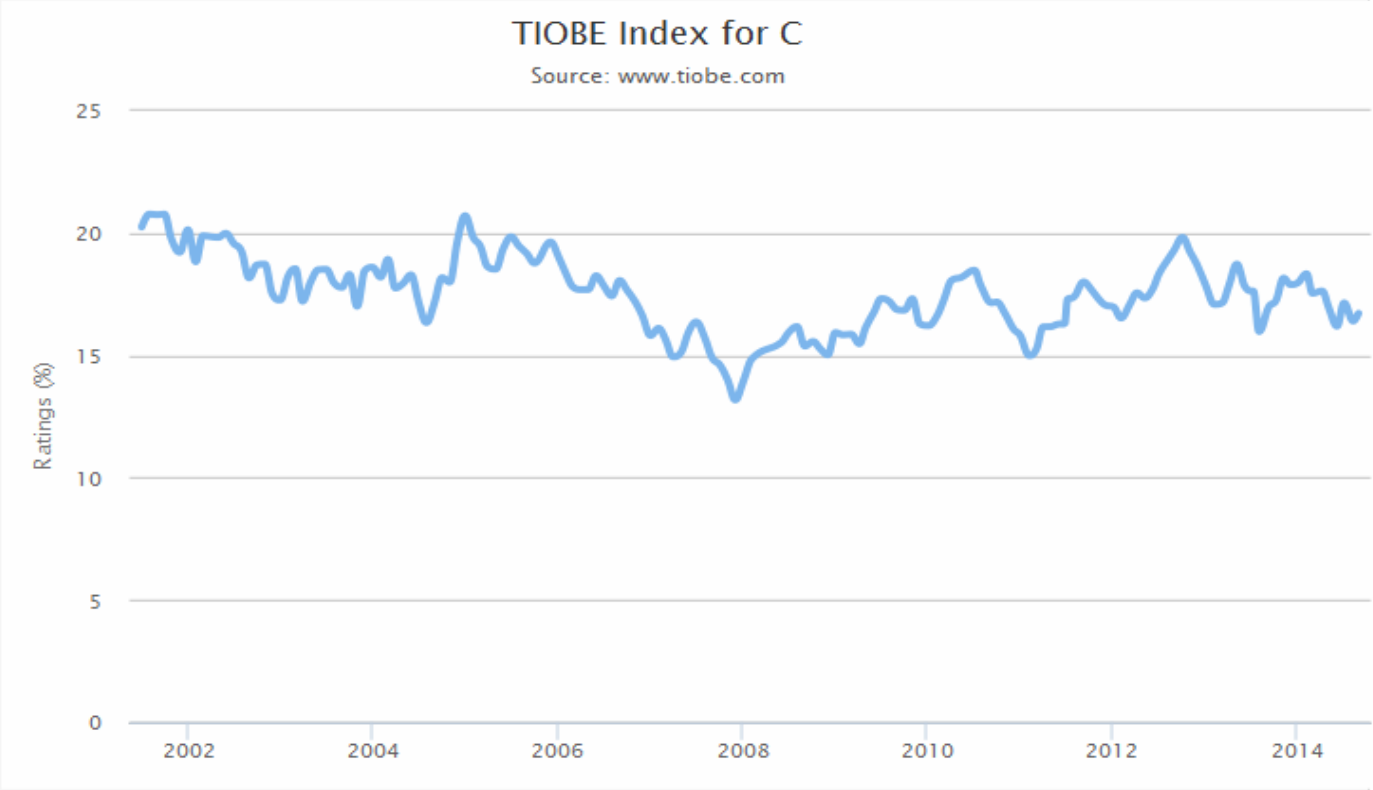
The C Programming Language

Some information about C:

Highest Position (since 1984): #1 in Sep 2014

Lowest Position (since 1984): #2 in Aug 2013

Language of the Year: 2008



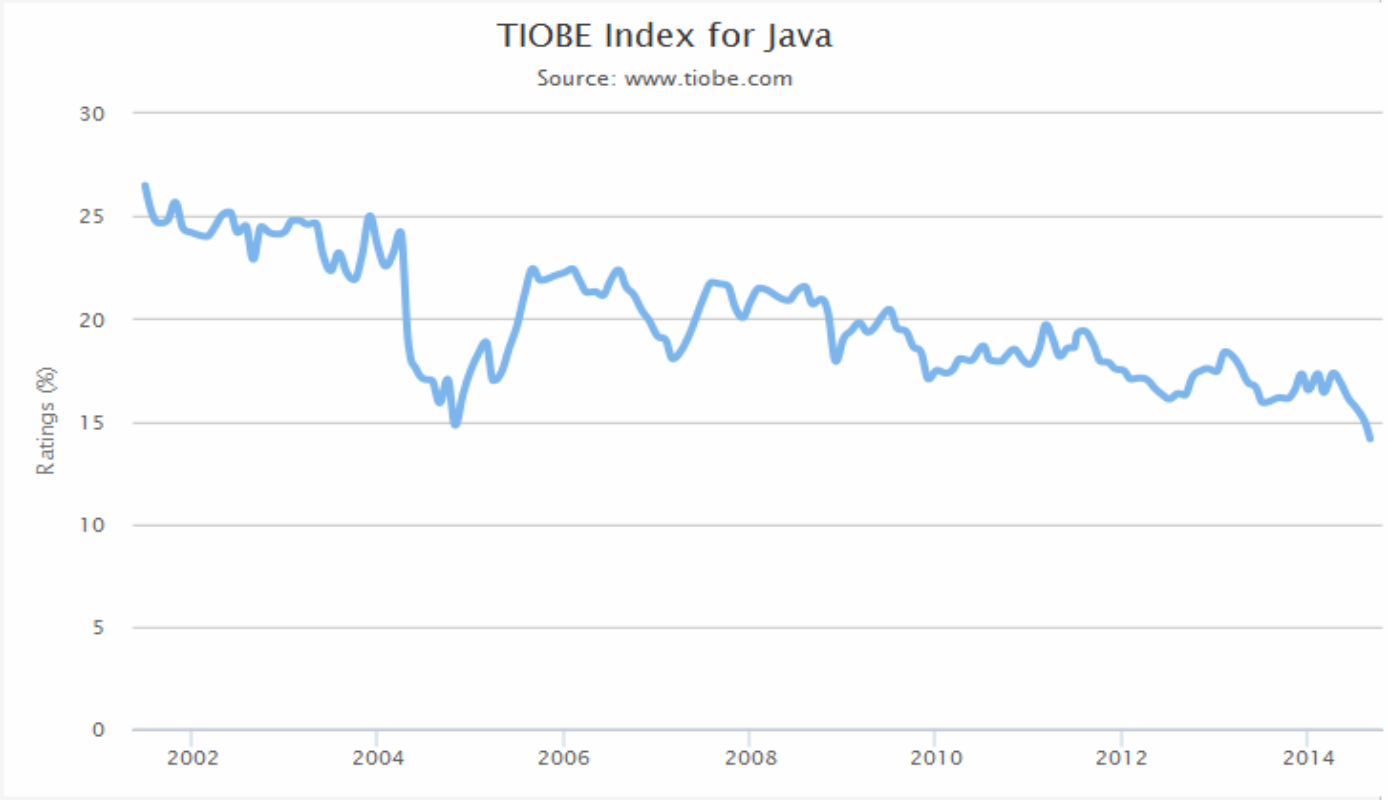
The Java Programming Language

Some information about Java:

Highest Position (since 1995): #1 in Aug 2013

Lowest Position (since 1995): #2 in Sep 2014

Language of the Year: 2005



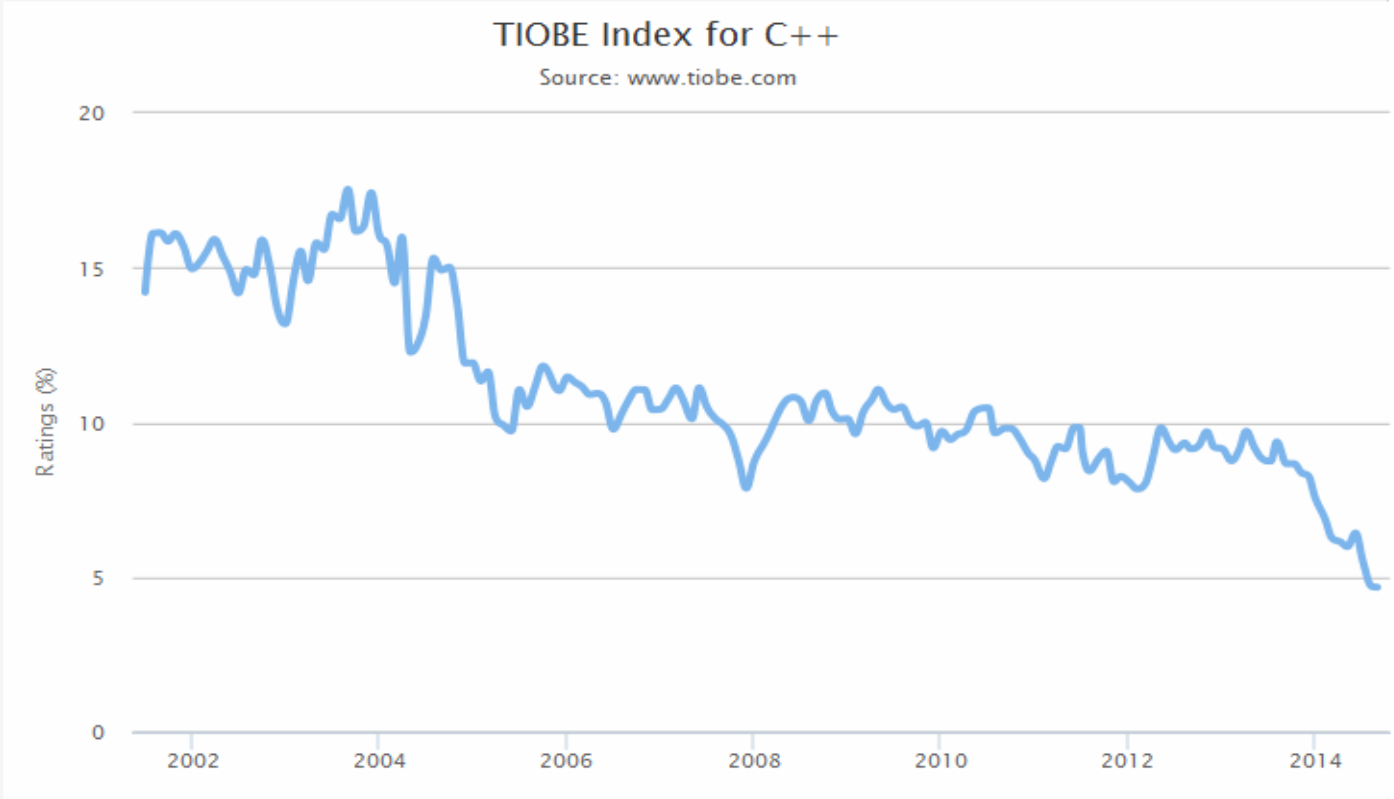
The C++ Programming Language

Some information about C++:

Highest Position (since 1984): #3 in Sep 2013

Lowest Position (since 1984): #5 in Feb 2008

Language of the Year: 2003



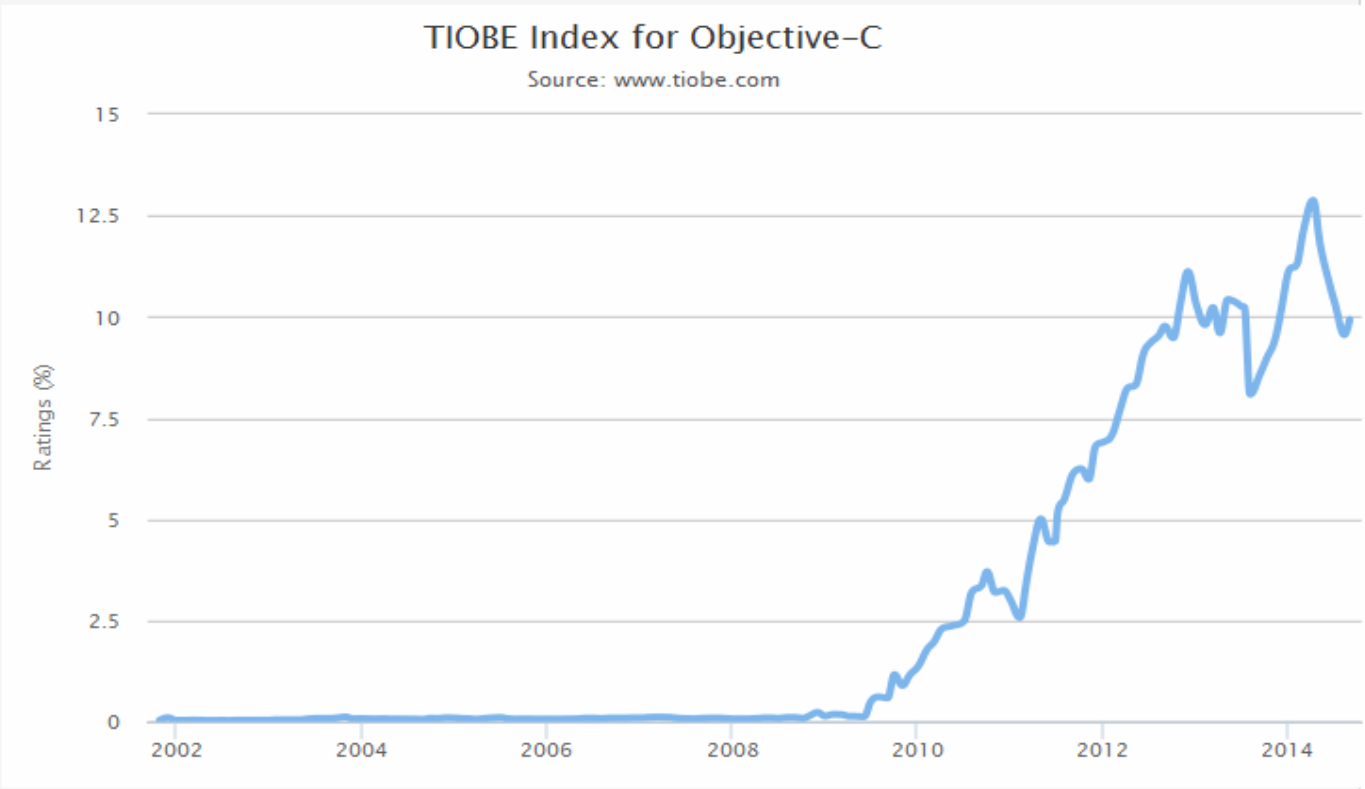
The Objective-C Programming Language

Some information about Objective-C:

Highest Position (since 2001): #3 in Sep 2014

Lowest Position (since 2001): #59 in Dec 2007

Language of the Year: 2011, 2012



Trendy Skills

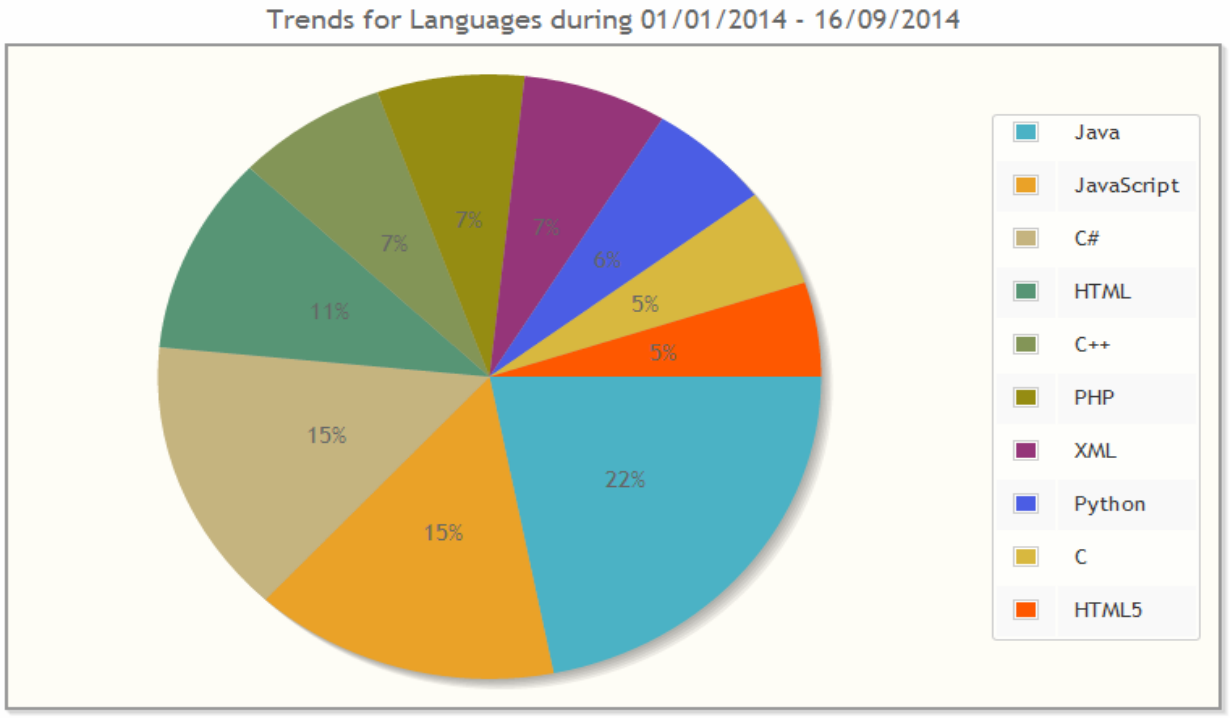
Extracting Skills that employers seek in the IT industry

Range 01/01/2014 - 16/09/2014

Search by Keyword Category

Tags Languages ×

Go Reset



Trendy Skills

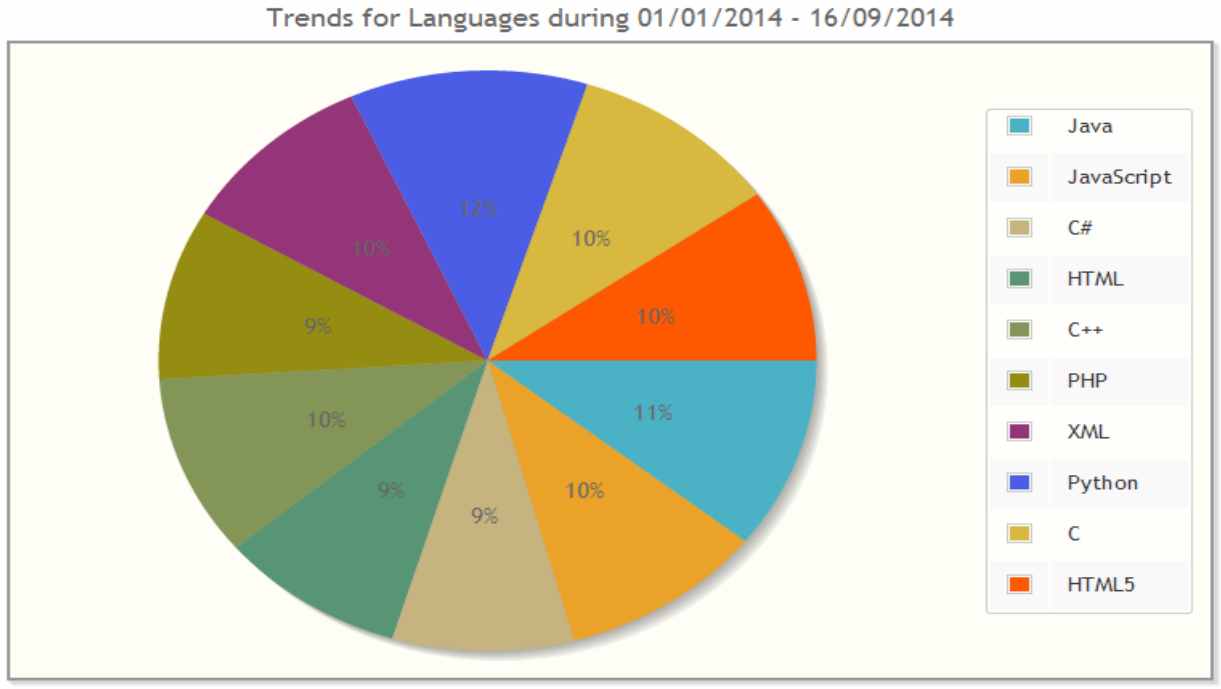
Extracting Skills that employers seek in the IT industry

Range 01/01/2014 - 16/09/2014

Search by Keyword Category

Tags Languages ×

Go Reset



Absolute | % Total | % Category | Salary ($)

Μεταγλωττιστής και Συνδέτης (Compiler & Linker)

- **Η C είναι *μεταγλωττισμένη* γλώσσα, με στόχο να είναι αναγνώσιμη και κατανοητή.**

- **'Compiler': πρόγραμμα μετατροπής αναγνώσιμο → εκτελέσιμο από τον Η/Υ**

- **Compile: C αρχεία κειμένου → .object file(s)**
- **Linking: object file(s) → .executable file**

Διαδικασία αποσφαλμάτωσης (Debugging)

- **'bug' = σφάλμα**
- **'debug' = εύρεση και διόρθωση σφαλμάτων**

Δύο είδη προγραμματιστικών σφαλμάτων:

Συντακτικά σφάλματα: **Σφάλματα που οφείλονται σε παραβίαση των συντακτικών κανόνων (π.χ. λανθασμένη γραφή μίας εντολής). Ανιχνεύονται από το μεταγλωττιστή κατά το χρόνο μεταγλώττισης και αναφέρονται υπό μορφή λίστας (οπότε μπορούν να διορθωθούν προτού τρέξει το πρόγραμμα).**

Σημασιολογικά σφάλματα (semantic errors): **Σφάλματα που οφείλονται σε εσφαλμένη σχεδίαση της λύσης του προβλήματος (π.χ. διαίρεση με το μηδέν). Τα σφάλματα αυτής της κατηγορίας δεν αναγνωρίζονται από το μεταγλωττιστή, εμφανίζονται αργότερα στο χρόνο εκτέλεσης και είναι δυσκολότερα στη διόρθωση καθώς απαιτούνται μεταβολές στη σχεδίαση και ακολούθως στον κώδικα).**

Debug: Εύρεση και διόρθωση σφαλμάτων

Κώδικας C → Μεταγλώττιση → Εκτέλεση, επανάληψη

- **Γίνεται η μεταγλώττιση;**
  - **ΟΧΙ→ ο μεταγλωττιστής αναφέρει τους λόγους. Τροποποίησε τον κώδικα και δοκίμασε εκ νέου**
- **ΝΑΙ. Λειτουργεί σωστά;**
  - **ΟΧΙ→ ανάλυσε, τροποποίησε, δοκίμασε εκ νέου**
- **ΝΑΙ. Αλλά δουλεύει πάντοτε σωστά;**

Μπορείς να αποδείξεις ότι η ανάλυση είναι εύρωστη;

Ένα απλό πρόγραμμα σε C

```
/* ************************************************
  *  This program prints out the sentence "This is a test."   *
  ************************************************ */
#include<stdio.h>

void main()
{
    printf("This is a test.\n");
}
```

Ένα απλό πρόγραμμα σε C

```
/*
  This program prints out the sentence "This is a test."
 */
#include<stdio.h>

void main ()
{
    printf("This is a test.\n");
}
```

Το **Σχόλιο (comment)** είναι κείμενο ανάμεσα σε /* και */. Να χρησιμοποιείτε συχνά σχόλια για να επεξηγείτε το πρόγραμμά σας και τα τμήματά του.

Ο μεταγλωττιστής δε λαμβάνει υπόψη τα σχόλια.

Ένα απλό πρόγραμμα σε C

#include σημαίνει "διάβασε και από αυτό το αρχείο"

```
/*
   This program prints out the sentence "This is a test."
*/
#include<stdio.h>

void main ()
{
   printf("This is a test.\n");
}
```

stdio είναι ένα αρχείο (library) που παρέχει "πρότυπες συναρτήσεις εισόδου/εξόδου" (όπως η *printf*)

Αρχείο με κατάληξη .h ονομάζεται αρχείο κεφαλίδας

Ένα απλό πρόγραμμα σε C

```
/*
   This program prints out the sentence "This is a test."
*/
#include<stdio.h>

void main ()
{
   printf("This is a test.\n");
}
```

Η εκτέλεση του προγράμματος αρχίζει πάντοτε από τη συνάρτηση *main()*.

ΟΛΑ τα προγράμματα σε C ή C++ έχουν συνάρτηση *main()*, η οποία συνήθως καλεί άλλες συναρτήσεις.

Ένα απλό πρόγραμμα σε C

```
#include<stdio.h>

void main ()
{
   printf("This is a test.\n");
}
```

Όλες οι συναρτήσεις χρησιμοποιούν άγκιστρα για να σημειώσουν την αρχή και το τέλος της συνάρτησης.

Το τμήμα ανάμεσα στα άγκιστρα ονομάζεται σώμα (body) της συνάρτησης.

Ένα απλό πρόγραμμα σε C

```
#include<stdio.h>


void main ()
{
    printf("This is a test.\n");
}
```

Μία συνάρτηση είναι ένα σύνολο προτάσεων με ένα δεδομένο όνομα, π.χ. *main()*, *printf()*.

Η πρόταση αυτή καλεί τη συνάρτηση *printf()* για να τυπώσει το καθορισμένο κείμενο.
Τα ορίσματα εισόδου (input arguments) περικλείονται από παρενθέσεις και προσδιορίζουν το προς εκτύπωση κείμενο και τη μορφή με την οποία θα εκτυπωθεί.

Ένα απλό πρόγραμμα σε C

```
#include<stdio.h>

void main ()
{
    printf("This is a test.\n");
}
```

ΟΛΕΣ οι προτάσεις τελειώνουν με ερωτηματικό (semicolon) ; το οποίο ονομάζεται **σύμβολο τερματισμού πρότασης.**

Λεπτομέρειες της *printf()*:

\n σημαίνει 'μετακινήσου στην επόμενη γραμμή'. Ονομάζεται **ακολουθία διαφυγής (escape sequence).**

Η έξοδος στην οθόνη είναι:

```
>This is a test.

>
```

Σύνταξη σχολίων

Τα σχόλια πρέπει να χρησιμοποιούνται αφειδώς γιατί καταστούν τον κώδικα ευανάγνωστο και συνεισφέρουν στην επεξήγηση δυσνόητων σημείων.

```
/* Το /* σχόλιο */ αυτό είναι λανθασμένο */

   /* Το σχόλιο αυτό χρησιμοποιεί σωστή σύνταξη */
                                                  /*
                  Ομοίως
και
αυτό.
                  */
```

Σύνταξη σχολίων

```
/* Το σχόλιο /* αυτό */ είναι λανθασμένο */
```

σχόλιο

κώδικας;

Να ευθυγραμμίζετε τα σύμβολα των σχολίων και να μην τοποθετείτε ποτέ σχόλια μέσα σε σχόλια (φώλιασμα, nesting) γιατί μερικοί μεταγλωττιστές θα μπερδευτούν.

Παρατηρήσεις:

- **Η γλώσσα C διαχωρίζει τα κεφαλαία γράμματα από τα μικρά (case sensitive). Η εντολή *Printf()* ΔΕΝ είναι ίδια με την *printf()*. Όλες οι εντολές στη C γράφονται με μικρά γράμματα!**
- **Η σωστή στηλοθεσία είναι πολύ σημαντική καθώς καθιστά τον κώδικα ευανάγνωστο.**
- **Να γράφετε πάντοτε σχόλια στα προγράμματά σας.**
- **Η εντολή *printf()* ανήκει στις *μορφοποιούμενες συναρτήσεις εισόδου–εξόδου*. Ονομάζεται μορφοποιούμενη γιατί δίνει τη δυνατότητα στο χρήστη να μορφοποιήσει την έξοδό της, δυνάμενη να εκτυπώσει μεταβλητές διαφόρων τύπων και με διάφορους τρόπους, χρησιμοποιώντας κατάλληλα σύμβολα. Δυαδική της *printf()* είναι η *scanf()*, η οποία λαμβάνει πληροφορία από την είσοδο (πληκτρολόγιο).**

Παρατηρήσεις (συνέχεια):

- **Η πρόταση `printf( "This is a test.\n" );` καλεί την *printf()* για να τυπωθεί το καθορισμένο κείμενο. Τα *ορίσματα εισόδου* (input arguments) περικλείονται από παρενθέσεις και προσδιορίζουν το προς εκτύπωση κείμενο και τη μορφή με την οποία θα εκτυπωθεί. Τέλος, το σύμβολο \n, που ανήκει στις *ακολουθίες διαφυγής*, σημαίνει «μετακινήσου στην επόμενη γραμμή». Λεπτομερής περιγραφή της λειτουργίας των συναρτήσεων εισόδου – εξόδου δίνεται στη συνέχεια.**
- **Πέραν της *include*, μία σημαντική οδηγία προς τον προεπεξεργαστή είναι η *define*, η οποία αντιστοιχίζει ένα όνομα με μία σταθερά ή με μία σειρά χαρακτήρων. Οποτεδήποτε εμφανίζεται το όνομα μέσα στον κώδικα, αντικαθίσταται αυτόματα με την τιμή της σταθεράς ή τη συμβολοσειρά.**

Για παράδειγμα, εάν χρησιμοποιηθεί η λέξη TRUE στη θέση της τιμής 1 και η λέξη FALSE στη θέση της τιμής 0, θα δοθούν δύο *define* ως εξής:

```
#define TRUE 1
#define FALSE 0
```

Εάν αντικατασταθεί ολόκληρη φράση, μπορεί να εμφανισθεί στην οθόνη με χρήση της *printf()*:

```
#define TITLOS  "Dpt of Informatics and Communications\n "
printf( TITLOS );
```

Το αποτέλεσμα είναι:

Dpt of Informatics and Communications

- **Μία συνηθισμένη χρήση της define είναι για τον καθορισμό του μεγέθους στοιχείων, όπως είναι η διάσταση ενός πίνακα, τα οποία μπορεί να αλλάξουν κατά την εκτέλεση του προγράμματος.**

Λεξιλόγιο της γλώσσας C

- *Δεσμευμένες λέξεις (reserved words)*
- *Λέξεις κλειδιά (keywords)*
- *Τελεστές (operators)*
- *Αναγνωριστές (identifiers)*

Δεσμευμένες λέξεις: πρέπει να αποφεύγεται η χρήση τους ως ονόματα

- **Ονόματα συναρτήσεων πρότυπης βιβλιοθήκης (runtime function names), όπως *printf()*, *abs()* κ.λ.π.**
- **Macro names. Είναι ονόματα που περιέχονται σε αρχεία κεφαλίδας για ορισμό μακροεντολών, π.χ. EOF, INT_MAX.**
- **Type names. Είναι ονόματα τύπων σε ορισμένα αρχεία κεφαλίδας, π.χ. time_t, va_list.**
- **Ονόματα εντολών προεπεξεργαστή (preprocessor). Είναι ονόματα που χρησιμοποιεί προεπεξεργαστής της C και έχουν προκαθορισμένη σημασία, π.χ. *include*, *define*.**
- **Ονόματα που αρχίζουν με το χαρακτήρα υπογράμμισης _ και έχουν δεύτερο χαρακτήρα τον ίδιο ή κεφαλαίο γράμμα, π.χ. _DATE_, _FILE_.**

Λέξεις κλειδιά: Λεκτικές μονάδες που μόνες τους ή με άλλες λεκτικές μονάδες χαρακτηρίσουν κάποια γλωσσική κατασκευή
Π.χ. **int**: αναπαριστά τον ακέραιο τύπο δεδομένων
if-else: χρησιμοποιούνται στον έλεγχο ροής προγράμματος

- Οι λέξεις κλειδιά, αν και είναι ένας περιορισμός των γλωσσών, αυξάνουν την αναγνωσιμότητα και αξιοπιστία των προγραμμάτων ενώ ταυτόχρονα επιταχύνουν τη διαδικασία της μεταγλώττισης.
- Λέξεις κλειδιά όπως **if**, **else**, **for**, **case**, **while**, **do** έχουν γίνει κοινά αποδεκτές, διευκολύνοντας την εκμάθηση των γλωσσών προγραμματισμού.

Λέξεις κλειδιά στην ANSI C:

auto	else	register	union
break	enum	return	unsigned
case	extern	short	void
char	float	signed	volatile
const	for	sizeof	while
continue	goto	static	
default	if	struct	
do	int	switch	
double	long	typedef	

Αναγνωριστές: **Λεκτικές μονάδες που κατασκευάζει ο προγραμματιστής. Αυτές οι λεκτικές μονάδες χρησιμοποιούνται συνήθως ως ονόματα που ο προγραμματιστής δίνει σε δικές του κατασκευές, όπως μεταβλητές, σταθερές, συναρτήσεις και δικούς του τύπους δεδομένων. Ένα όνομα προσδιορίζει μοναδιαία (uniquely identifies), από το σύνολο των κατασκευών του προγράμματος, την κατασκευή στην οποία αποδόθηκε, εξ ου και το όνομα *αναγνωριστής* (identifier).**

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- **Αποφύγετε ονόματα ενός χαρακτήρα, όπως *i*, *j*, *x*, *y* (εκτός από ειδικές περιπτώσεις που θα εξετασθούν αργότερα).**
- **Χρησιμοποιείτε εκφραστικά ονόματα. Συγκεκριμένα:**

  - ονομάστε τη μεταβλητή που αναπαριστά την ταχύτητα ως velocity και τη μέγιστη τιμή της max_velocity ή maxVelocity.

  - ονομάστε τη συνάρτηση που εμφανίζει τα λάθη στην οθόνη *display_error* ή *displayError*.
- **Για καλύτερη αναγνωσιμότητα των μεταβλητών που αποτελούνται από δύο ή περισσότερες λέξεις αποφασίστε αν θα χρησιμοποιείτε τη μορφή *display_error* ή τη μορφή *displayError*. Τηρείστε τη σύμβαση σε όλο το πρόγραμμα.**
- **Χρησιμοποιείστε μικρά γράμματα για ονόματα μεταβλητών.**