



Πανεπιστήμιο Κρήτης
Τμήμα Επιστήμης Υπολογιστών
www.csd.uoc.gr



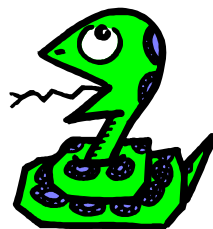
Εργαστήριο Υπηρεσιών
Μετασχηματισμού
www.tsl.gr

HY-100 Εισαγωγή στην Επιστήμη Υπολογιστών

<http://efront.tsl.gr>

Δεύτερη Διάλεξη

Επιστήμη Υπολογιστών, Αλγόριθμοι, Προγραμματισμός

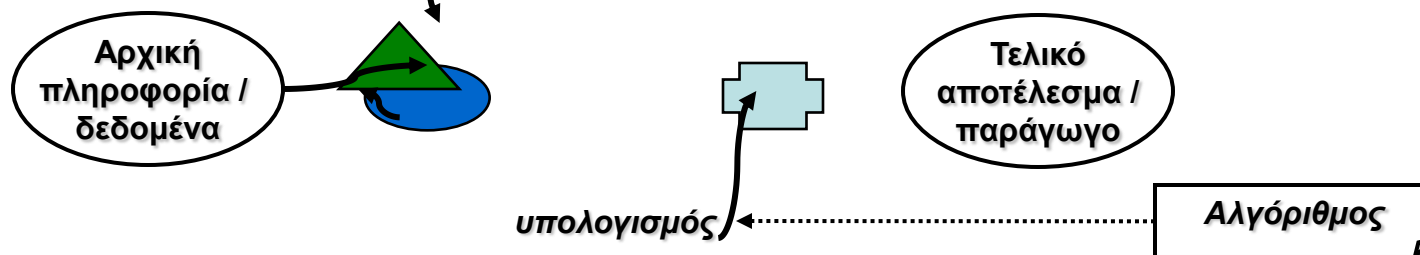


Διδάσκων: Νικολάου Χρήστος

Χειμερινό εξάμηνο 2009-2010
Δευτέρα, 28/09/2009

Τι είναι η επιστήμη υπολογιστών

- Η μελέτη της *πληροφορίας*, και των τρόπων *αναπαράστασης*, *οργάνωσης*, *διαχείρισης*, και *μετασχηματισμού* για την *επίλυση συγκεκριμένων προβλημάτων*.
- Η διαχείριση και ο μετασχηματισμός γίνονται μέσω βημάτων της διαδικασίας *υπολογισμού* (computation), η οποία και περιγράφεται με *αλγόριθμους* (algorithms).





Περί αλγορίθμων (1/5)

- *Περιγραφή διαδικασίας που εκτελείται με πεπερασμένα διαδοχικά βήματα για την επίλυση ενός συγκεκριμένου προβλήματος.*
- Όλοι έχουμε εμπειρία σχετικά με:
 - οδηγίες χρήσης / επιδιόρθωσης / σύνθεσης κάποιου προϊόντος.
 - συνταγές μαγειρικής, π.χ. «*Στην κουζίνα ολοταχώς™*», «*Συνταγές της Βέφας™*», κλπ, κλπ.



Περί αλγορίθμων (2/5)

- Απόσπασμα οδηγιών – *εγκατάσταση ψυγείου*

<p>...</p> <ul style="list-style-type: none">• Μην τοποθετήσετε τίποτε στη συσκευή.• Κλείστε όλες τις πόρτες της συσκευής.• Βάλετε το καλώδιο στην πρίζα.• Ανοίξτε το διακόπτη της συντήρησης.• Ανοίξτε το διακόπτη της ψύξης.	<ul style="list-style-type: none">• Μόνο όταν η θερμοκρασία ψύξης δείξει -18 βαθμούς μπορείτε να τοποθετήσετε τρόφιμα.• Μόνο όταν η θερμοκρασία συντήρησης δείξει 4 βαθμούς μπορείτε να τοποθετήσετε τρόφιμα. <p>...</p>
--	---

- Φαίνονται ξεκάθαρα τα συγκεκριμένα **ακολουθιακά βήματα**, το ένα μετά το άλλο, όπως υποδεικνύονται από τις οδηγίες.
- Ορίζονται και **δύο συνθήκες** για ορθή εγκατάσταση της συσκευής, που ουσιαστικά ορίζουν πότε μπορεί να χρησιμοποιηθεί. Οι συνθήκες αυτές είναι: **Θερμοκρασία συντήρησης: 4** και **Θερμοκρασία ψύξης: -18**



Περί αλγορίθμων (3/5)

- Απόσπασμα οδηγιών – *εκκίνηση αυτοκινήτου*

<p>...</p> <ul style="list-style-type: none">• Βάζετε την σχέση του κιβωτίου στην ένδειξη «νεκρά».• Ελέγχετε το χειρόφρενο το οποίο πρέπει να είναι ενεργοποιημένο.• Βάζετε το κλειδί στη θέση του διακόπτη της μηχανής.• Πατάτε το συμπλέκτη.• Γυρίζετε το διακόπτη στη θέση ON και <i>περιμένετε μέχρι</i> να σβήσει η ένδειξη IMMO.	<ul style="list-style-type: none">• Γυρίζετε το κλειδί στη θέση START και τον κρατάτε εκεί <i>έως ότου</i> ξεκινήσει η μηχανή ή <i>έως το πολύ 8 δευτερόλεπτα</i>. Έπειτα αφήνετε απλώς το κλειδί.• <i>Εάν δεν</i> εκκινήσει η μηχανή, επαναλάβετε. <i>Εάν μετά από 5 επαναλήψεις δεν</i> εκκινήσει η μηχανή, απευθυνθείτε σε επίσημο συνεργείο. <p>...</p>
--	--

- Και πάλι έχουμε πολλά βήματα, αλλά τώρα εμφανίζονται αρκετές συνθήκες που απαιτούν έλεγχο και ανάλογη ενέργεια.
- Για πρώτη φορά εμφανίζεται επανάληψη ενέργειας, και μάλιστα με συνθήκη τερματισμού: **επαναλήψεις: 5 ή η μηχανή εκκίνησε κανονικά**



Περί αλγορίθμων (4/5)

- Παρατηρούμε ότι σε αυτές τις περιγραφές εμφανίζονται συχνά οδηγίες για ενέργειες που πρέπει να γίνονται:
 - **Ακολουθιακά**
 - Βήματα της διαδικασίας ακολουθούν σε διάταξη το ένα μετά το άλλο
 - **Υπό συνθήκη**
 - Βήματα και υπο-διαδικασίες ακολουθούνται μόνο εάν ισχύει μία συνθήκη που πρέπει να ελέγχεται εάν αληθεύει
 - **Επαναληπτικά**
 - Βήματα και υπο-διαδικασίες επαναλαμβάνονται μέχρι κάποια συνθήκη λήξης της επανάληψης



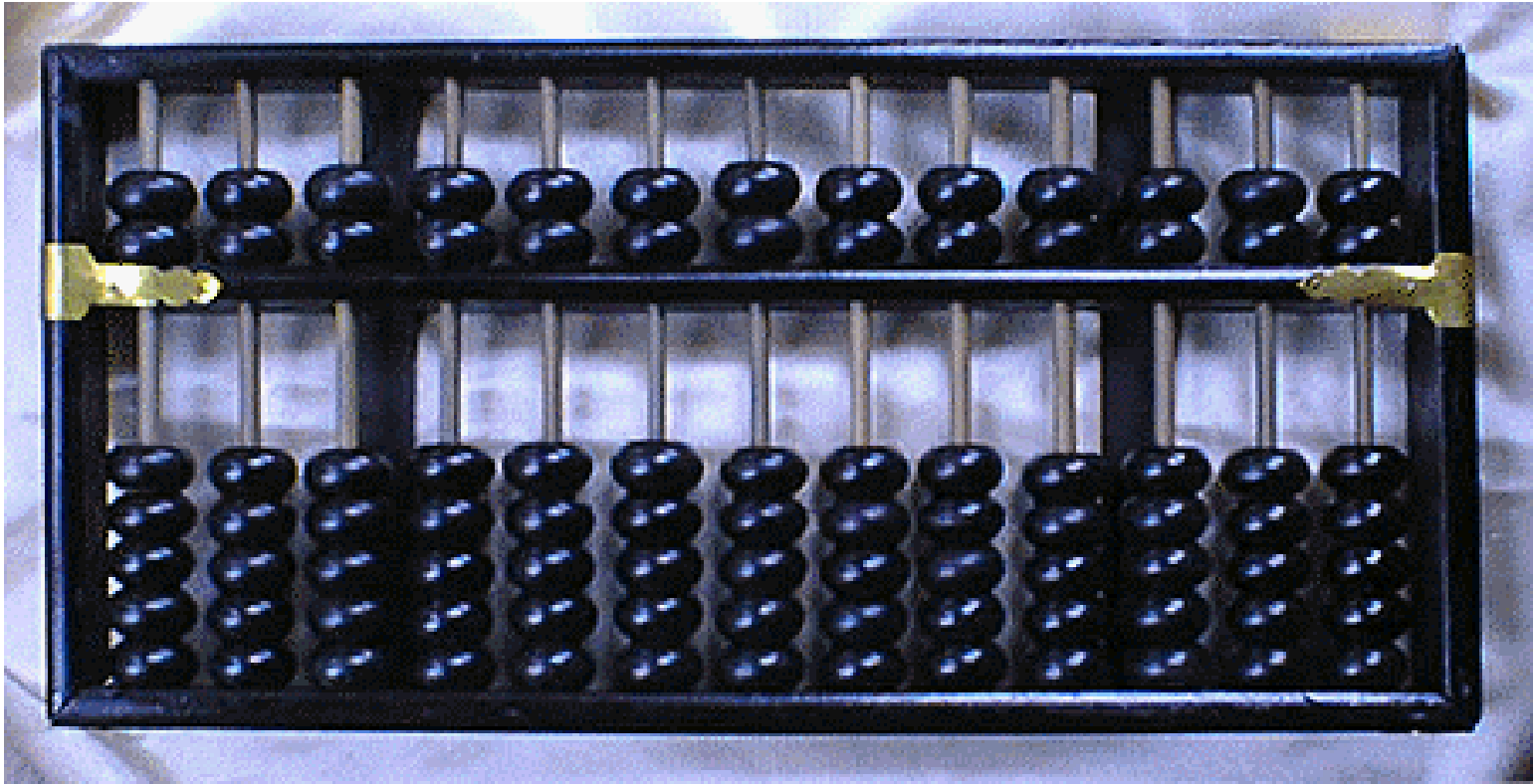


Περί αλγορίθμων (5/5)

- Δίνουμε έναν πιο λεπτομερή ορισμό για το τι είναι αλγόριθμος:
 - Ο αλγόριθμος είναι μία πεπερασμένη αλληλουχία
 - από σαφείς και μη διφορούμενες εντολές
 - που όλες δύνανται να εκτελεστούν και να περατωθούν
 - ο οποίος όταν εκτελείται παράγει το επιθυμητό αποτέλεσμα
 - ☒ *και οδηγεί σε τερματισμό της όλης διαδικασίας σε πεπερασμένο χρόνο – λάθος.*



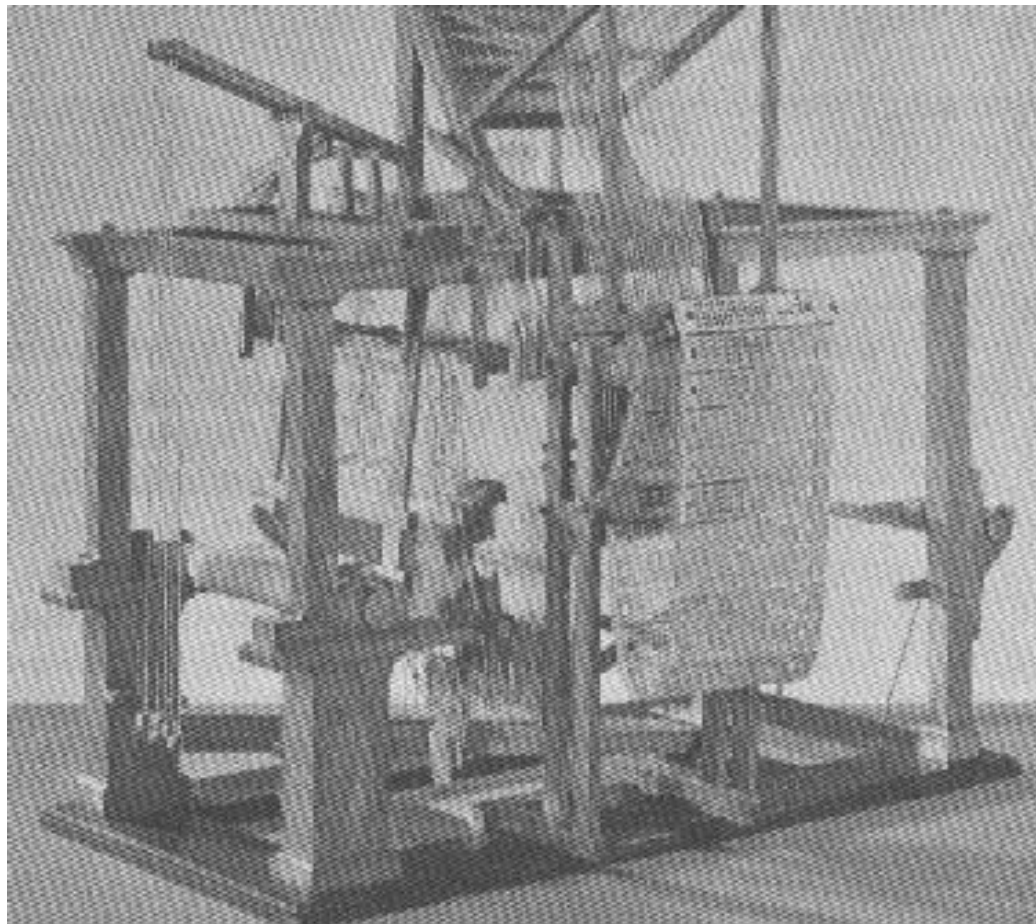
Abacus (5th Century B.C.)



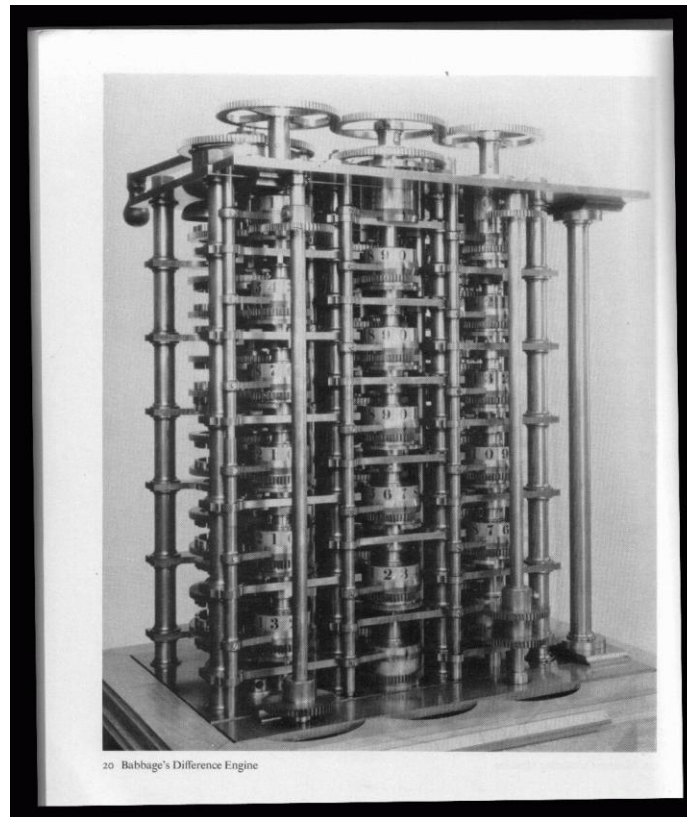
Pascaline (1643)



Jacquard's Loom (1805)



Babbage's Difference Engine (1832)



Hollerith's Census Machine (1890)



Harvard's Mark I (1947)



Microcomputers (1972-2009)





Επεξεργασία Πληροφορίας

- Ξεκίνησε πριν 2500 χρόνια με την εφεύρεση της γραφής στην αρχαία Μεσοποταμία και την Ελλάδα.
 - Γραφή: χρήση συμβόλων / γραμμάτων στα καθημερινά κείμενα
 - Μαθηματικά: χρήση συμβόλων για να αναπαραστήσουν αφηρημένα συστήματα ιδεών και αλληλουχίες συνεπαγωγών
 - Επιστήμη Υπολογιστών: αυτόματη επεξεργασία συμβόλων





Η Υπόθεση Church-Turing

- Εάν μπορούμε να σχεδιάσουμε ένα αλγόριθμο για να λύσει ένα πρόβλημα, τότε μπορούμε (τουλάχιστον θεωρητικά) να κατασκευάσουμε ένα πραγματικό υπολογιστή που *αυτοματοποιεί* τη λύση.



Μια Γενικευμένη Μηχανή (Universal Machine)

- Μερικοί υπολογιστές είναι πολύ εξειδικευμένοι, ενώ άλλοι είναι γενικής χρήσης.
- Ένας υπολογιστής γενικής χρήσης μπορεί (κατ' αρχήν) να *αυτοματοποιήσει τη λύση κάθε αλγοριθμικού προβλήματος*, δηλαδή κάθε προβλήματος για το οποίο μπορεί να διατυπωθεί μια λύση μέσω ενός αλγορίθμου.





Αλγόριθμοι και Προγράμματα

- Οι αλγόριθμοι είναι αφηρημένες ιδέες, φτιαγμένοι από τη στόφα της ανθρώπινης σκέψης...
- Τα *προγράμματα* εκφράζουν αλγορίθμους σε *συγκεκριμένες γλώσσες προγραμματισμού*, όπως:
 - Basic
 - Pascal
 - FORTRAN
 - C
 - C++
 - Java
 - Python
 - Lisp





Hardware and Software



- *Hardware*

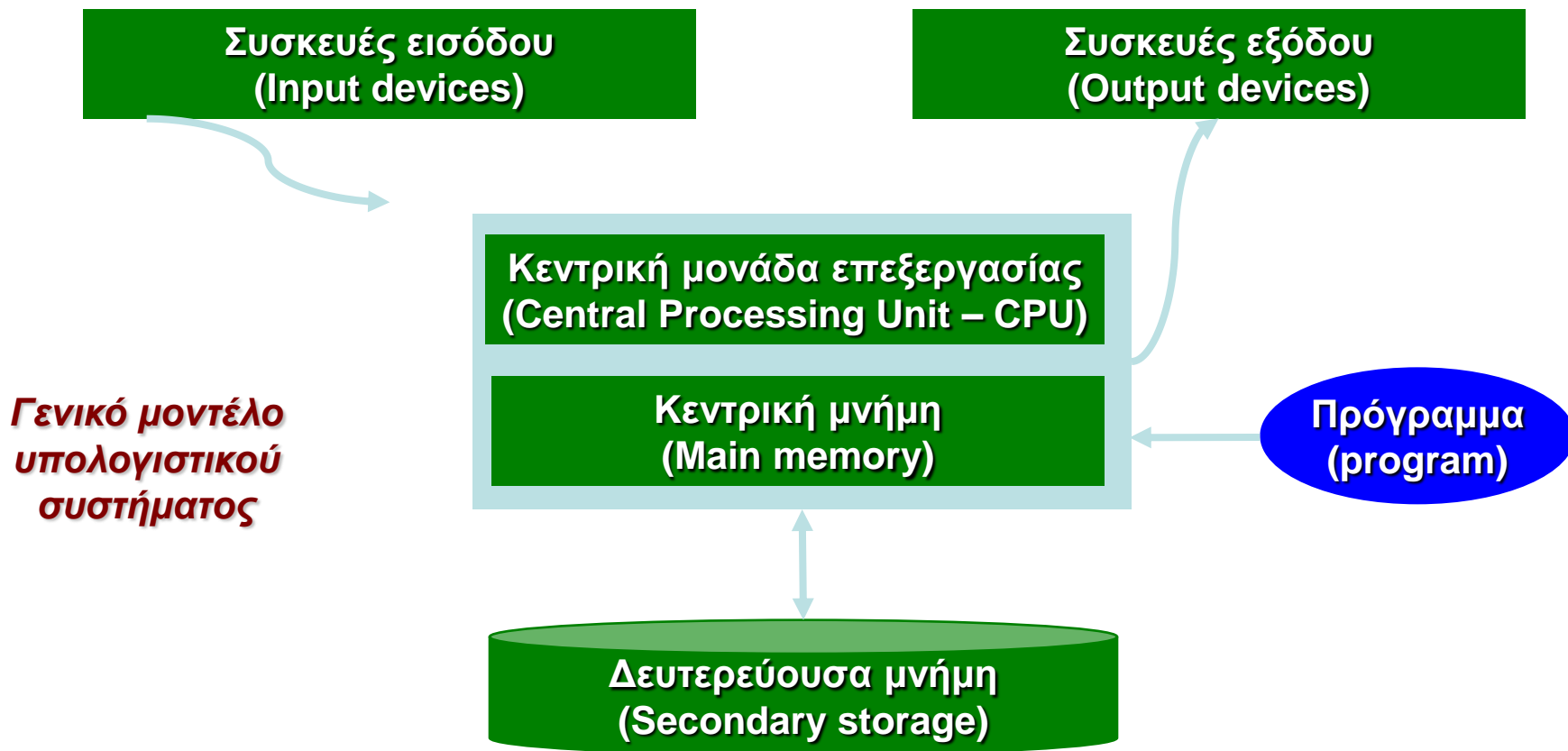
- αποτελείται από συσκευές – input/output devices, processor, memory storage – που είναι απαραίτητες για να τρέξουν τα προγράμματά μας.

- *Software*

- αποτελείται από προγράμματα για τη επίλυση προβλημάτων, συμπεριλαμβανομένων και αυτών που επεξεργάζονται άλλα προγράμματα.



Ο Υπολογιστής



Τι κάνει ένας Επιστήμονας των Υπολογιστών;

- Σκοπός του μαθήματος HY-100 είναι να αρχίσετε να σκέπτεστε ως επιστήμονες των υπολογιστών.
- Ο τρόπος αυτός σκέψης συνδυάζει στοιχεία σκέψης από τα **Μαθηματικά**, την Επιστήμη του **Μηχανικού** και της **Φυσικής**.
- Όπως οι μαθηματικοί, οι επιστήμονες των υπολογιστών χρησιμοποιούν τυπικές **γλώσσες** (formal languages) για να εκφράσουν ιδέες, και συγκεκριμένα υπολογισμούς.
- Όπως οι μηχανικοί, σχεδιάζουν «πράγματα από hardware ή software», συναρμολογούν συνιστώμενα μέρη σε συστήματα και αξιολογούν διάφορες εναλλακτικές λύσεις.
- Ως φυσικοί επιστήμονες, παρατηρούν την συμπεριφορά πολύπλοκων συστημάτων, διαμορφώνουν υποθέσεις και ελέγχουν προβλέψεις.



Η Επίλυση Προβλημάτων

- Η πιο σημαντική δεξιότητα ενός Επιστήμονα των Υπολογιστών είναι η **επίλυση προβλημάτων**.
- Επίλυση προβλημάτων είναι η ικανότητα να διατυπώνει κανείς προβλήματα, να σκέπτεται δημιουργικά για την επίλυσή τους και να εκφράζει τη λύση καθαρά και με ακρίβεια.
- Από καταπληκτική σύμπτωση, όταν μαθαίνει κανείς να προγραμματίζει, έχει μια εξαιρετική ευκαιρία να ακονίσει τη δεξιότητά του να λύνει προβλήματα.



- *Προγραμματισμός* είναι η διαδικασία:
 - *Ανάλυσης* ενός προβλήματος
 - *Σχεδίασης* μιας αλγοριθμικής λύσης
 - *Κωδικοποίησης* της σχεδίασης σε μια συγκεκριμένη γλώσσα προγραμματισμού
 - *Ελέγχου* της ορθότητας της λύσης
 - *Συντήρησης* της λύσης (feedback και αλλαγή της σχεδίασης και του κώδικα)



Γλώσσες Προγραμματισμού (1/2)

- Η γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε λέγεται **Python**.
- Η Python είναι παράδειγμα γλώσσας προγραμματισμού **υψηλού επιπέδου**. Άλλες τέτοιες γλώσσες είναι οι C, C++, Perl και Java.
- Για να υπάρχουν γλώσσες προγραμματισμού υψηλού επιπέδου, θα πρέπει να υπάρχουν και γλώσσες προγραμματισμού **χαμηλού επιπέδου** ή όπως αλλιώς λέγονται «γλώσσες μηχανής» ή γλώσσες assembly.
- Χωρίς να είμαστε ιδιαίτερα ακριβείς, μπορούμε να πούμε ότι οι υπολογιστές μπορούν μόνο να εκτελέσουν προγράμματα γραμμένα σε γλώσσες χαμηλού επιπέδου.
- Άρα τα προγράμματα που είναι γραμμένα σε γλώσσες υψηλού επιπέδου πρέπει να υποστούν **μια επεξεργασία** πριν τρέξουν. Η επιπλέον επεξεργασία επιτελείται από άλλα προγράμματα που λέγονται **διερμηνευτές και μεταγλωττιστές**.

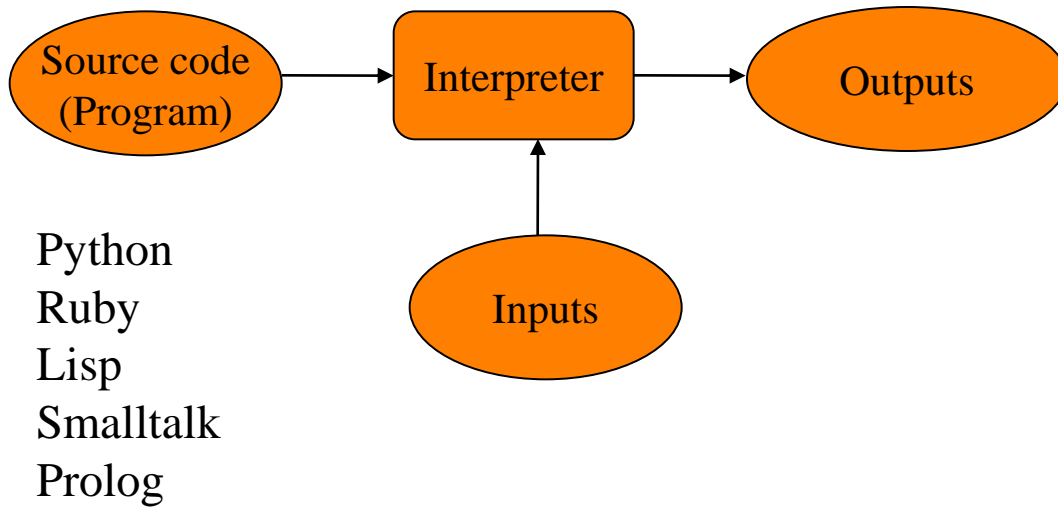


Γλώσσες Προγραμματισμού (2/2)

- Αυτή η επιπλέον επεξεργασία παίρνει βεβαίως χρόνο και αυτό είναι σημαντικό μειονέκτημα των γλωσσών υψηλού επιπέδου.
- Αλλά υπάρχουν τεράστια **πλεονεκτήματα**:
 - Είναι πολύ πιο **εύκολο** να προγραμματίσεις σε γλώσσα υψηλού επιπέδου.
 - Προγράμματα γραμμένα σε γλώσσες υψηλού επιπέδου **γράφονται γρηγορότερα**, είναι πιο **μικρά** και πιο **εύκολα** στην ανάγνωση (είναι πιο κοντά στους ανθρώπους).
 - Έχουν μεγαλύτερη πιθανότητα να είναι **σωστά**.
 - Επίσης τα προγράμματα αυτά εύκολα **μεταφέρονται** σε διαφορετικούς υπολογιστές, με καμία ή λίγες αλλαγές.
- Γι' αυτούς τους λόγους τα πιο πολλά προγράμματα γράφονται σε γλώσσες υψηλού επιπέδου και ελάχιστα σε γλώσσες χαμηλού επιπέδου.

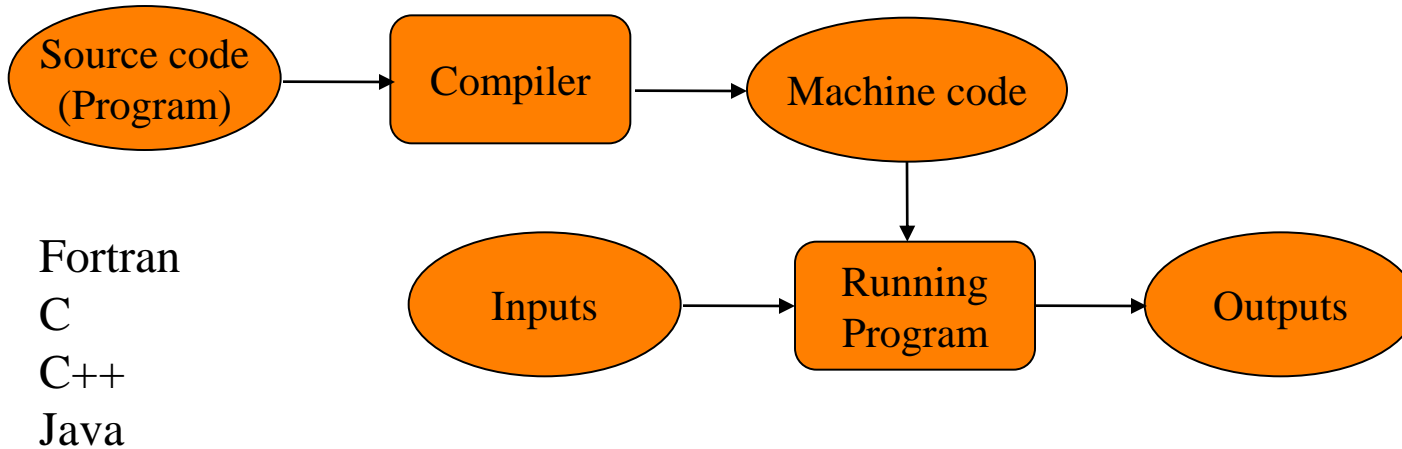


Διερμηνευτές (Interpreters)



- *Interpreted languages* translate a program during execution

Μεταγλωττιστές (Compilers)



- *Compiled languages* require a separate compilation step before program execution



01 111001 000001 1101 10 0111 010001 11 0110 010111 001000 011110 0101 11 0010 0001
0100 00111010 01011100 0001 1010 010111 11 0100 0101 1010 11 0101 1010 10010000 11101 0010 001111 10101
11101011101