Σπουδές και Εισαγωγή στον Προγραμματισμό

Φροντιστήριο

Αριστείδης Ηλίας

Το Τμήμα και οι Σπουδές

Κτίριο

- **Δημιουργήθηκε το 1979, άρχισε τη λειτουργία του το 1980**

Πρώτο Τμήμα Υπολογιστών

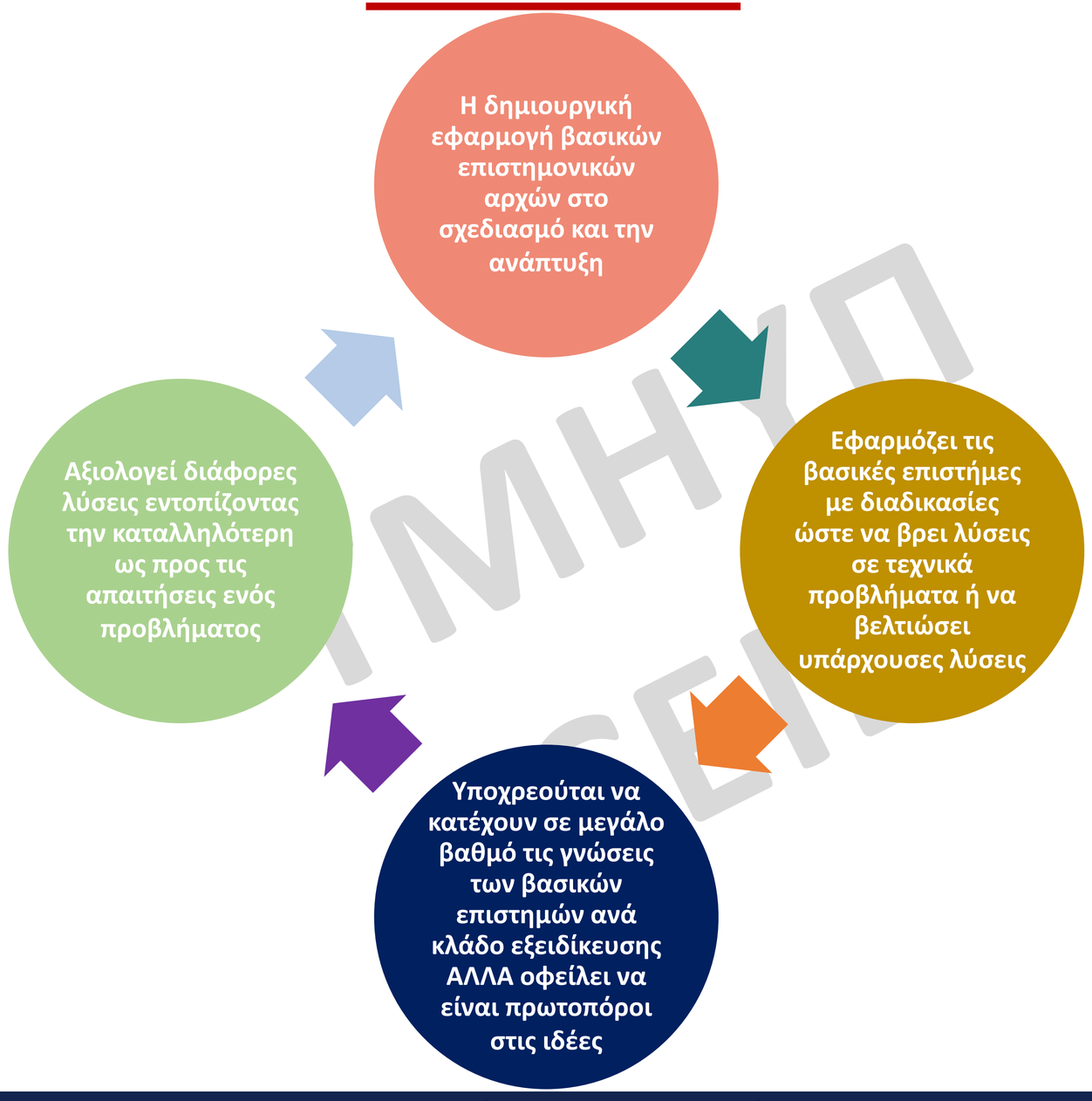
Οι πρώτοι των πρώτων

- **Αναγνωρισμένο στη χώρα και διεθνώς**

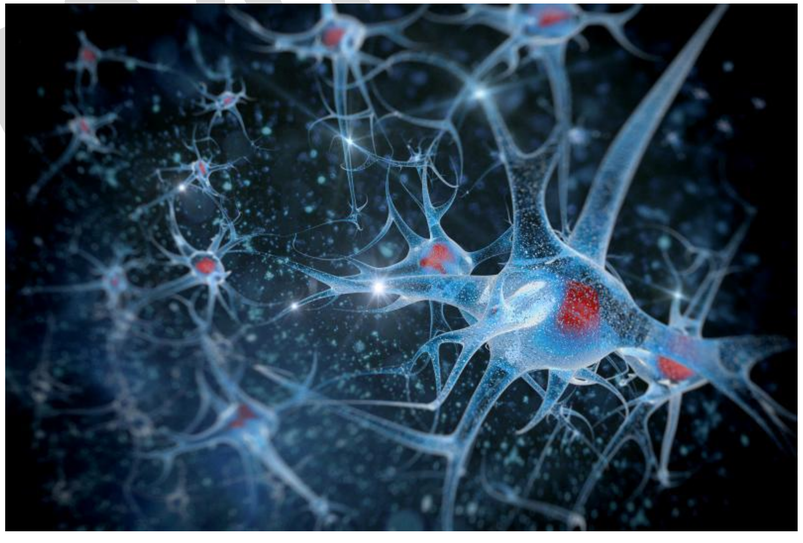
Τομείς


Εφαρμογών
και
Θεμελιώσεων
της Επιστήμης
των
Υπολογιστών
Λογικού των
Υπολογιστών
Υλικού και
Αρχιτεκτονικής
των
Υπολογιστών

Μηχανικός



Πώς θα περάσουμε το μάθημα;

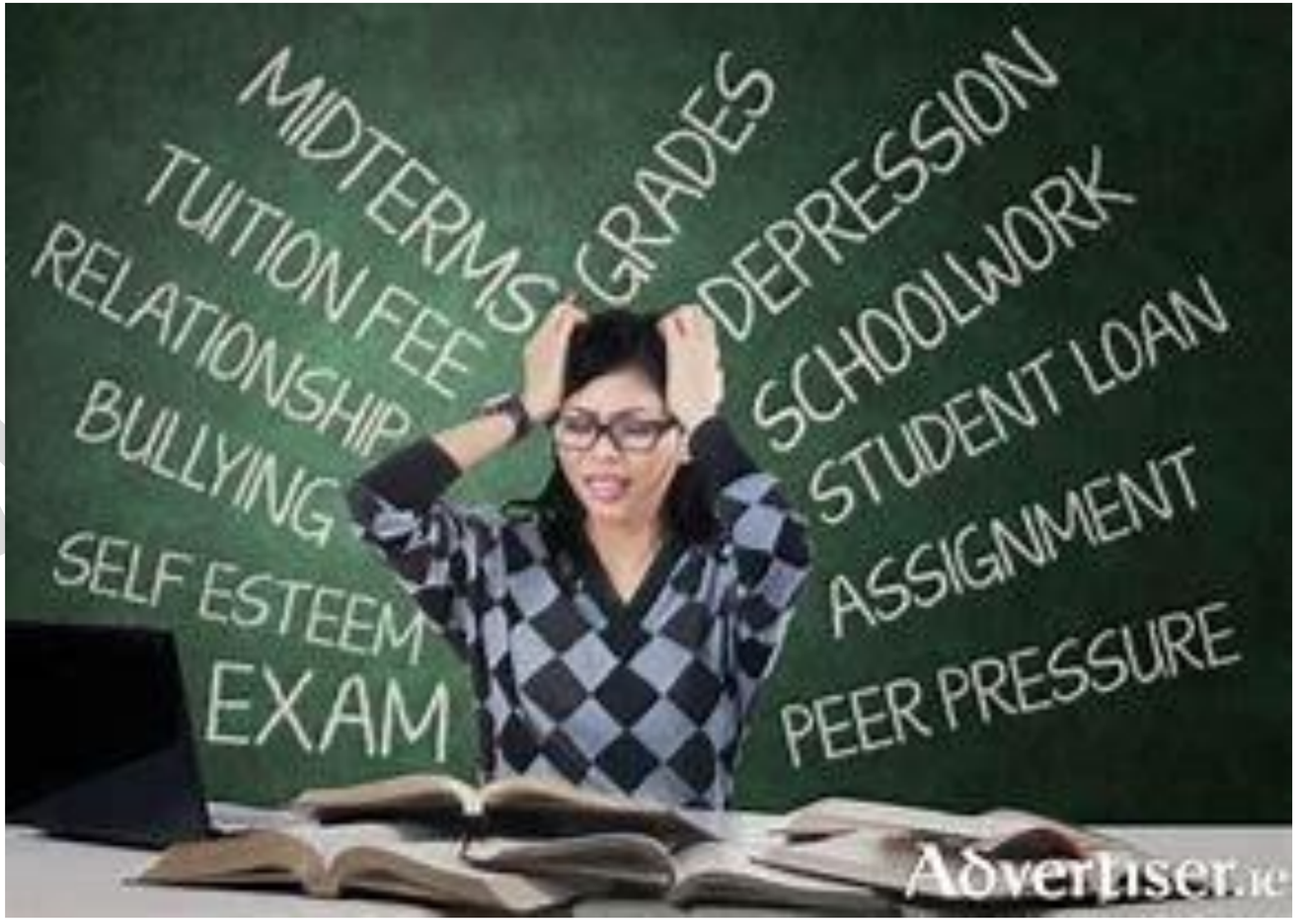


Πώς θα μάθω;

- **Μην εφησυχάσετε**

- **Δουλέψτε στο πρώτο έτος για να βάλετε τις βάσεις που οφείλετε στον εαυτό σας**

- **Παρακολουθείτε**
- **Μελετάτε (από που???)**
  - **Ξεχνάτε σελίδες/ παραγράφους/ φωτοτυπίες/ διαφάνειες**
  - **Η ύλη είναι τα πάντα**
- **Ερευνάτε**
- **Προετοιμάζεστε**
- **Το 30% είναι ο δάσκαλος**
- **Το 70% εσείς**



…Συμβουλές «Παλιών»…

Δεοντολογία

Το μάθημα αρχίζει και τέταρτο

ΜΗΝ ΠΕΡΙΜΕΝΕΤΕ να χτυπήσει το κουδούνι

«Ο καθηγητής είναι ο τελευταίος που μπαίνει στην αίθουσα»… παρόλα αυτά θα είμαστε εδώ πριν από εσάς

Μπορείτε να μπείτε ΑΛΛΑ από τις πίσω πόρτες και με ΗΣΥΧΙΑ!!!

Εάν αργήσετε πολύ ή χάσετε ώρες… ίσως ΔΕΝ ΕΧΕΙ ΕΝΝΟΙΑ ΝΑ ΜΠΕΙΤΕ…

Το μάθημα στο πανεπιστήμιο δεν είναι υποχρεωτικό να το παρακολουθείτε… ΔΕΝ ΕΧΕΙ νόημα όμως να έρχεστε για να σχολιάσετε τι φοράει ο ένας κι ο άλλος ή πως τα περάσατε χθες βράδυ στην Πάτρα

Πάμε στο μάθημα συνειδητοποιημένα γιατί θέλουμε να μάθουμε

…Συμβουλές «Παλιών»…

Δεοντολογία

Απουσίες στα εργαστήρια

Κρατάτε σημειώσεις όπου πιστεύετε ότι χρειάζονται, άρα χρειάζομαι τετράδιο μαζί

Παρακολουθείτε τα μαθήματα

Μελετάτε εβδομαδιαία χωρίς κενά

Ρωτάτε απορίες στο μάθημα… χωρίς προβληματισμό για τα σχόλια!!!

Μην ακούτε «φήμες» και μην εμπιστεύεστε τις «ομαδικές»…

Οι εξετάσεις στο πανεπιστήμιο

Ιανουάριος-Φεβρουάριος

- Μαθήματα Χειμερινού

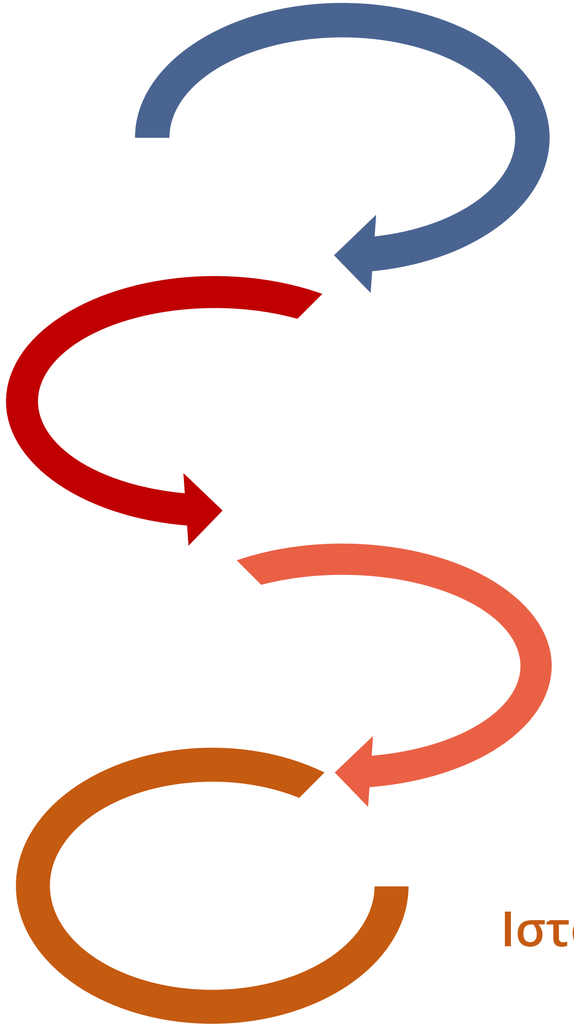
Ιούνιος

- Μαθήματα Εαρινού

Τέλος Αυγούστου- Σεπτέμβριος (επαναληπτική)

- Χειμερινού-Εαρινού

Ενημέρωση φοιτητών- Πως ενημερώνομαι;;;...



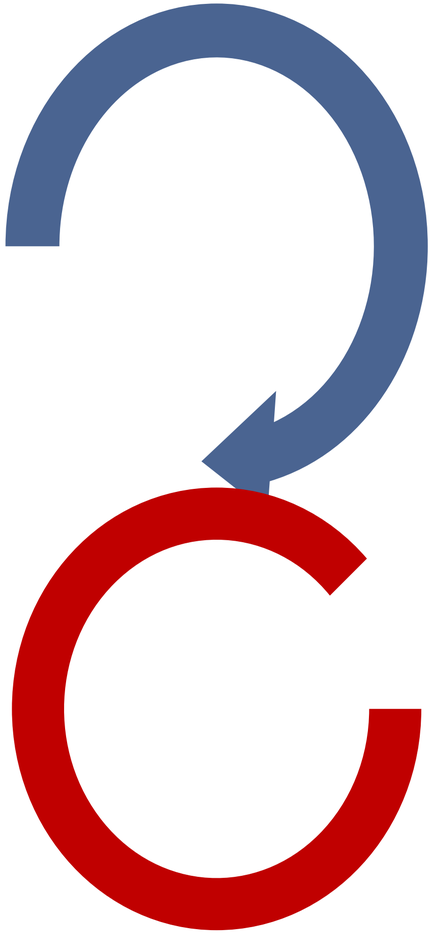
https://eclass.upatras.gr

Ανακοινώσεις τμήματος (www.ceid.upatras.gr)

Ηλεκτρονικό Ταχυδρομείο Τμήματος

Ιστότοπους μαθημάτων

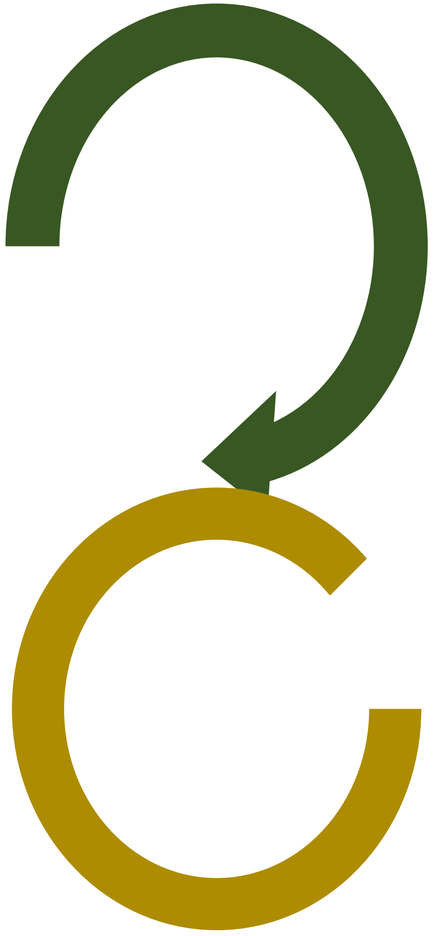
Παρακολουθώ τα «δρώμενα» του τμήματος και μαθημάτων



Progress για δηλώσεις μαθημάτων

ΕΥΔΟΞΟΣ για τα βιβλία

«Ανυπερθέτως» παρακολουθώ και δεν ξεχνώ τις προθεσμίες!!!



Πανεπιστημίου (up<AM>@ac.upatras.gr)

https://webmail.ac.upatras.gr/

https://www.upnet.gr/e-manuals/students-email/

Τμήματος (<username>@ceid.upatras.gr)

Καλή Σταδιοδρομία

Εργαστήριο Προγραμματισμού

Εμπέδωση και κατανόηση των αρχών και των μεθοδολογιών προγραμματισμού

- Υπολογιστικό Κέντρο

- Η συμμετοχή σας στο εργαστήριο είναι **υποχρεωτική** καθώς είναι ο μόνος τρόπος για να μάθετε προγραμματισμό
- Περισσότερες από δύο (2) αδικαιολόγητες απουσίες σας οδηγούν σε αποτυχημένη παρακολούθηση του (STRONG FAIL)
- Αυτό σας στερεί το δικαίωμα συμμετοχής στην Εξέταση Εργαστηρίου για την 1η ακαδημαϊκή χρονιά που δηλώσατε το μάθημα, δηλαδή στην εξεταστική του Φεβρουαρίου 2025
- Αυτόματα αυτό σας οδηγεί σε Βαθμό Εργαστηρίου(ΒΕ) μηδέν (0) και υποχρέωση εκτέλεσης του εργαστηρίου σε επόμενη ακαδημαϊκή χρονιά

Οργάνωση…

- Το εργαστήριο διεξάγεται κατά ομάδες, αλλά ατομικά
- Θα υπάρξει σχετική ανακοίνωση που θα αναρτηθεί στο eclass
- Εκτός από την ομάδα η ανακοίνωση θα σας ενημερώνει και για την θέση σας στο Υπολογιστικό κέντρο
- Οι θέσεις θα είναι αριθμημένες στο διάγραμμα που θα αναρτηθεί
- **Οι παρουσίες στο εργαστήριο λαμβάνονται με βάσει την ομάδα και τις θέσεις οπότε είναι δική σας ευθύνη να κάθεστε στην θέση σας σε κάθε εργαστήριο**

…Οργάνωση

- Οι κατάλογοι με τις ομάδες (ώρες συμμετοχής και παρακολούθησης) θα αναρτηθούν μετά το πέρας μόλις ολοκληρωθούν οι εγγραφές

- **Για την ημερομηνία εκτέλεσης του πρώτου εργαστηρίου θα ενημερωθείτε από την σχετική ανακοίνωση στο eclass**

Προετοιμασία…

- Σε κάθε εργαστήριο θα εκτελείτε ένα σύνολο από ασκήσεις
- Πρέπει να έχετε προετοιμαστεί κατάλληλα για την εκτέλεση τους μελετώντας
- Σημαντικό είναι να έχετε παρακολουθήσει τις διαλέξεις και το φροντιστήριο
- Είναι απαραίτητα για να μάθετε και για την αποτελεσματική εκτέλεση του κάθε εργαστηρίου
- Αν για οποιοδήποτε λόγο δεν καταφέρετε να συμμετέχετε στη διάλεξη ή/και φροντιστήριο θα πρέπει να προετοιμαστείτε με δική σας ευθύνη πριν από την προσέλευση σας στο εργαστήριο
- Αν βρεθείτε απροετοίμαστος είναι στην διακριτική ευχέρεια του διδάσκοντα να σας θεωρήσει απόντα/απούσα.

...Προετοιμασία

- Στο εργαστήριο θα έρθετε αφού έχετε μελετήσει το υλικό που σας έχει δοθεί και έχετε δουλέψει με αυτό στο βαθμό που μπορείτε στο σπίτι σας ή πίνοντας καφέ με τους συμφοιτητές σας (σας το συνιστούμε καθώς είναι πιο αποτελεσματικό)
- Στο εργαστήριο θα λύσετε τις απορίες σας πάνω στις ασκήσεις αλλά και στις έννοιες που αυτό σας αναγκάζει να χρησιμοποιήσετε
- Κατά τη διάρκεια του εργαστηρίου θα εκτελείτε τις εργαστηριακές ασκήσεις επικουρούμενοι από τους επιβλέποντες του εργαστηρίου

1ο Παραδοτέο

- Λίγο πριν την ολοκλήρωση του χρόνου του κάθε εργαστηρίου θα σας ζητείται να παραδώσετε στο eclass υλικό που δημιουργήσατε κατά την εκτέλεση του εργαστηρίου (πρώτο παραδοτέο)
- Για την διαδικασία υποβολής του πρώτου αυτού παραδοτέου στο eclass θα ενημερωθείτε από σχετική ανακοίνωση
- Για το τι θα παραδώσετε σε κάθε εργαστήριο θα ενημερώνεστε την ώρα του αντίστοιχου εργαστηρίου
- Τα πρώτα αυτά παραδοτέα σας δεν λαμβάνονται υπόψη στη διαμόρφωση του ΒΕ

Τελικό Παραδοτέο

- Την εργαστηριακή άσκηση (σύνολο ασκήσεων) θα πρέπει να ολοκληρώσετε και στη συνέχεια να υποβάλετε το τελικό παραδοτέο, αργότερα σε χρόνο που θα σας ζητηθεί και πάντα πριν την οριζόμενη κάθε φορά καταληκτική ημερομηνία παράδοσης
- Το παραδοτέο είναι ατομική εργασία
- Ενώ ενθαρρύνεται η συνεργασία και η ανταλλαγή απόψεων με τους συμφοιτητές σας, απαγορεύονται αυστηρά η λογοκλοπή και η αντιγραφή κώδικα, τμήματος ή συνόλου, καθώς και η οικειοποίηση κώδικα που αναπτύχθηκε από άλλους
- Αυτό είναι κάτι που ελέγχουμε ηλεκτρονικά με το turnItIn.

Βαθμολογία…

Βαθμός Μαθήματος = 60% Βαθμός της τελικής γραπτής εξέτασης του μαθήματος + 40% ΒΕ

- Ο ΒΕ συμμετέχει στην διαμόρφωση του τελικού βαθμού του μαθήματος για την τρέχουσα ακαδημαϊκή χρονιά και την επομένη
- Μετά διαγράφεται χωρίς αυτό να σημαίνει υποχρέωση επανάληψης του εργαστηρίου
- Στην περίπτωση αυτή θα πρέπει να επαναλάβετε την εξέταση εργαστηρίου
- Βαθμός Εργαστηρίου (ΒΕ) διαμορφώνεται από τον βαθμό της αξιολόγησης της συμμετοχής σας στο εργαστήριο (Βαθμός Αξιολόγησης Συμμετοχής στο Εργαστήριο – ΒΑΣΕ) και τον βαθμό εξέτασης εργαστηρίου (Βαθμό Εξέτασης Εργαστηρίου – ΒΕΕ) σύμφωνα με τον παρακάτω τύπο

ΒΕ = 60% ΒΑΣΕ + 40% ΒΕΕ

…Βαθμολογία…

- Βαθμός Εξέτασης Εργαστηρίου – ΒΕΕ, ο βαθμός αυτός είναι ο βαθμός που θα λάβετε στην Εξέταση Εργαστηρίου
- Βαθμός Αξιολόγησης Συμμετοχής στο Εργαστήριο – ΒΑΣΕ, προκύπτει από:
  - ✓ Α) την αξιολόγηση των παραδοτέων σας, όπου τα τελευταία έχουν διπλάσια συμμετοχή από τα πρώτα, και
  - ✓ Β) την γενικότερη παρουσία σας μέσα στον χώρο του εργαστηρίου

ΒΑΣΕ = 60% [10 - (αριθμός FAIL παραδοτέων κατηγορίας Α * 1) - (αριθμός FAIL παραδοτέων κατηγορίας Β * 2)] + 40% Βαθμός γενικότερης παρουσίας την ώρα του εργαστηρίου.

- Ο παραπάνω τύπος δεν ισχύει για την περίπτωση που το σύνολο των παραδοτέων που δεν υποβάλατε και αυτών που χαρακτηρίστηκαν ως FAIL είναι ίσο ή μεγαλύτερο από το 40% των παραδοτέων (οπότε ΒΑΣΕ=0)
- Για την κατηγορία του κάθε παραδοτέου (Α ή Β) θα ενημερώνεστε με την ανακοίνωση του παραδοτέου

…Βαθμολογία

- **PASS/FAIL Παραδοτέου**
  - ✓ **Κάθε παραδοτέο σας χαρακτηρίζεται με PASS ή FAIL. Ένα παραδοτέο χαρακτηρίζεται ως Αποδεκτό (PASS) όταν είναι υλοποιημένο σωστά τουλάχιστο κατά το 50%**

- **Εξέταση Εργαστηρίου**
  - ✓ **Η εξέταση Εργαστηρίου λαμβάνει χώρα μετά την ολοκλήρωση των εβδομαδιαίων εργαστηρίων και πριν από τις τελικές εξετάσεις του εξαμήνου**
  - ✓ **Για την εξέταση αυτή θα ενημερωθείτε από σχετική ανακοίνωση**
  - ✓ **Στην Εξέταση Εργαστηρίου μπορείτε να συμμετέχετε αν δεν έχετε λάβει STRONG FAIL στο εργαστήριο**

Αρχές Προγραμματισμού

Στόχος

- Κατανόηση της έννοιας του προβλήματος
- Σαφήνεια στη διατύπωση
- Αναλυτική σκέψη στην προσέγγιση

- Πρόβλημα είναι μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση δεν είναι γνωστή, ούτε προφανής
- Για την αντιμετώπιση κάθε προβλήματος πρέπει προηγουμένως να έχει προηγηθεί η κατανόησή του. Αποτελεί συνάρτηση δυο παραγόντων:
  - Σωστή διατύπωση
  - Σωστή ερμηνεία από αυτόν που θα το επιλύσει
- Με τον όρο δομή προβλήματος αναφερόμαστε στα συστατικά του μέρη, στα επιμέρους τμήματα που το αποτελούν καθώς επίσης και στον τρόπο που αυτά συνδέονται μεταξύ τους.

- Ο Νίκος και η Δήμητρα είναι παντρεμένοι
  - Πρώτη ερμηνεία: Ο Νίκος και η Δήμητρα είναι παντρεμένοι μεταξύ τους.
  - Δεύτερη ερμηνεία: Ο Νίκος είναι παντρεμένος και η Δήμητρα είναι παντρεμένη, αλλά όχι μεταξύ τους.

- Διαγραμματική ανάλυση
- Η σωστή επίλυση ενός προβλήματος προϋποθέτει τον επακριβή προσδιορισμό των δεδομένων που παρέχει το πρόβλημα. Απαιτεί επίσης την λεπτομερειακή καταγραφή των ζητουμένων που αναμένονται σαν αποτελέσματα της επίλυσης του προβλήματος
- Τα στάδια αντιμετώπισης εντός προβλήματος είναι τρία
  - Κατανόηση: σωστή και πλήρης αποσαφήνιση των δεδομένων και ζητουμένων του προβλήματος
  - Ανάλυση: το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα
  - Επίλυση: λύση του προβλήματος, με επίλυση των αρχικών προβλημάτων

Κατηγορίες Προβλημάτων

- Τα στάδια αντιμετώπισης εντός προβλήματος είναι τρία
  - Κατανόηση: σωστή και πλήρης αποσαφήνιση των δεδομένων και ζητουμένων του προβλήματος
  - Ανάλυση: το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα
  - Επίλυση: λύση του προβλήματος, με επίλυση των αρχικών προβλημάτων
- Με κριτήριο της δυνατότητα επίλυσης ενός προβλήματος
  - Επιλύσιμα. Τα προβλήματα που η λύση τους είναι γνωστή και έχει διατυπωθεί ή η συνάφειά τους με ήδη επιλυμένα προβλήματα επιτρέπει να θεωρούμε βέβαιη την επίλυσή τους
  - Ανοικτά. Η λύση τους δεν έχει ακόμη βρεθεί αλλά δεν έχει αποδειχθεί ότι δεν λύνονται
  - Άλυτα. Έχει αποδειχθεί ότι δεν επιλύονται
- Με κριτήριο το βαθμό δόμησης ενός προβλήματος
  - Δομημένα. Η επίλυση προέρχεται από μια αυτοματοποιημένη διαδικασία
  - Ημιδομημένα. Η λύση τους επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων αφήνοντας στον ανθρώπινο παράγοντα περιθώρια επιλογής της
  - Αδόμητα. Οι λύσεις τους δεν μπορούν να δομηθούν λη δεν έχει διερευνηθεί η δυνατότητα δόμησής τους
- Με κριτήριο το είδος επίλυσης ενός προβλήματος
  - Απόφασης. Το πρόβλημα πρέπει πάρει μια απόφαση συνήθως σε μια ερώτηση ναι/όχι
  - Υπολογιστικά. Η λύση τους απαιτεί τη διενέργεια πολύπλοκων υπολογισμών
  - Βελτιστοποίησης. Το πρόβλημα επιζητά το βέλτιστο αποτέλεσμα για συγκεκριμένο σετ δεδομένων

Προβλήματα και Υπολογιστής

- Οι υπολογιστές δρουν επικουρικά στην ανθρώπινη δραστηριότητα, δεν έχουν τις δυνατότητες του ανθρώπινου εγκεφάλου.
- Σε αυτούς αναθέτουμε την επίλυση προβλημάτων λόγω:
  - Πολυπλοκότητας υπολογισμών
  - Επαναληπτικότητα διαδικασιών
  - Ταχύτητα εκτέλεσης πράξεων
  - Μεγάλο πλήθος δεδομένων
- Ο υπολογιστής μπορεί να πραγματοποιήσει τις εξής τρεις λειτουργίες
  - Πρόσθεση
  - Σύγκριση (λογικές πράξεις)
  - Μεταφορά δεδομένων

Προβλήματα και Ανάλυση

Στόχος

- Κατανόηση της έννοιας του προβλήματος
- Εισαγωγή στους Αλγορίθμους

Στόχος

- Κατανόηση της έννοιας του προβλήματος
- Σαφήνεια στη διατύπωση
- Αναλυτική σκέψη στην προσέγγιση

- Πρόβλημα είναι μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση δεν είναι γνωστή, ούτε προφανής
- Για την αντιμετώπιση κάθε προβλήματος πρέπει προηγουμένως να έχει προηγηθεί η κατανόησή του. Αποτελεί συνάρτηση δυο παραγόντων:
  - Σωστή διατύπωση
  - Σωστή ερμηνεία από αυτόν που θα το επιλύσει
- Με τον όρο δομή προβλήματος αναφερόμαστε στα συστατικά του μέρη, στα επιμέρους τμήματα που το αποτελούν καθώς επίσης και στον τρόπο που αυτά συνδέονται μεταξύ τους.

- Ο Νίκος και η Δήμητρα είναι παντρεμένοι
  - Πρώτη ερμηνεία: Ο Νίκος και η Δήμητρα είναι παντρεμένοι μεταξύ τους.
  - Δεύτερη ερμηνεία: Ο Νίκος είναι παντρεμένος και η Δήμητρα είναι παντρεμένη, αλλά όχι μεταξύ τους.

- Διαγραμματική ανάλυση
- Η σωστή επίλυση ενός προβλήματος προϋποθέτει τον επακριβή προσδιορισμό των δεδομένων που παρέχει το πρόβλημα. Απαιτεί επίσης την λεπτομερειακή καταγραφή των ζητουμένων που αναμένονται σαν αποτελέσματα της επίλυσης του προβλήματος
- Τα στάδια αντιμετώπισης εντός προβλήματος είναι τρία
  - Κατανόηση: σωστή και πλήρης αποσαφήνιση των δεδομένων και ζητουμένων του προβλήματος
  - Ανάλυση: το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα
  - Επίλυση: λύση του προβλήματος, με επίλυση των αρχικών προβλημάτων

Κατηγορίες Προβλημάτων

- Τα στάδια αντιμετώπισης εντός προβλήματος είναι τρία
  - Κατανόηση: σωστή και πλήρης αποσαφήνιση των δεδομένων και ζητουμένων του προβλήματος
  - Ανάλυση: το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα
  - Επίλυση: λύση του προβλήματος, με επίλυση των αρχικών προβλημάτων
- Με κριτήριο της δυνατότητα επίλυσης ενός προβλήματος
  - Επιλύσιμα. Τα προβλήματα που η λύση τους είναι γνωστή και έχει διατυπωθεί ή η συνάφειά τους με ήδη επιλυμένα προβλήματα επιτρέπει να θεωρούμε βέβαιη την επίλυσή τους
  - Ανοικτά. Η λύση τους δεν έχει ακόμη βρεθεί αλλά δεν έχει αποδειχθεί ότι δεν λύνονται
  - Άλυτα. Έχει αποδειχθεί ότι δεν επιλύονται
- Με κριτήριο το βαθμό δόμησης ενός προβλήματος
  - Δομημένα. Η επίλυση προέρχεται από μια αυτοματοποιημένη διαδικασία
  - Ημιδομημένα. Η λύση τους επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων αφήνοντας στον ανθρώπινο παράγοντα περιθώρια επιλογής της
  - Αδόμητα. Οι λύσεις τους δεν μπορούν να δομηθούν λη δεν έχει διερευνηθεί η δυνατότητα δόμησής τους
- Με κριτήριο το είδος επίλυσης ενός προβλήματος
  - Απόφασης. Το πρόβλημα πρέπει πάρει μια απόφαση συνήθως σε μια ερώτηση ναι/όχι
  - Υπολογιστικά. Η λύση τους απαιτεί τη διενέργεια πολύπλοκων υπολογισμών
  - Βελτιστοποίησης. Το πρόβλημα επιζητά το βέλτιστο αποτέλεσμα για συγκεκριμένο σετ δεδομένων

Προβλήματα και Υπολογιστής

- Οι υπολογιστές δρουν επικουρικά στην ανθρώπινη δραστηριότητα, δεν έχουν τις δυνατότητες του ανθρώπινου εγκεφάλου.
- Σε αυτούς αναθέτουμε την επίλυση προβλημάτων λόγω:
  - Πολυπλοκότητας υπολογισμών
  - Επαναληπτικότητα διαδικασιών
  - Ταχύτητα εκτέλεσης πράξεων
  - Μεγάλο πλήθος δεδομένων
- Ο υπολογιστής μπορεί να πραγματοποιήσει τις εξής τρεις λειτουργίες
  - Πρόσθεση
  - Σύγκριση (λογικές πράξεις)
  - Μεταφορά δεδομένων

Γλώσσες Προγραμματισμού…

- Για να γραφτεί ένα πρόγραμμα για έναν υπολογιστή, χρειάζεται μια γλώσσα προγραμματισμού
- Οι γλώσσες προγραμματισμού είναι σύνολα από προκαθορισμένες λέξεις οι οποίες συνδυάζονται σε προγράμματα σύμφωνα με προκαθορισμένους κανόνες (σύνταξη)
- Με το πέρασμα των χρόνων οι γλώσσες προγραμματισμού εξελίχθηκαν από γλώσσες μηχανής σε γλώσσες υψηλού επιπέδου

…Γλώσσες Προγραμματισμού…

- Την εποχή των πρώτων υπολογιστών, οι μοναδικές γλώσσες προγραμματισμού που υπήρχαν ήταν οι γλώσσες μηχανής (machine languages)

- Κάθε υπολογιστής είχε τη δική του γλώσσα μηχανής, η οποία αποτελούνταν από σειρές μηδενικών και άσσων

- Το επόμενο βήμα στην εξέλιξη του προγραμματισμού πραγματοποιήθηκε με την ιδέα της αντικατάστασης του δυαδικού κώδικα που χρησιμοποιούνταν για τις εντολές και τις διευθύνσεις με σύμβολα ή μνημονικούς κωδικούς (mnemonics)

- Συμβολικές γλώσσες (symbolic languages) ή assembly

Η μοναδική γλώσσα που καταλαβαίνει ένας υπολογιστής είναι η γλώσσα μηχανής.

…Γλώσσες Προγραμματισμού…

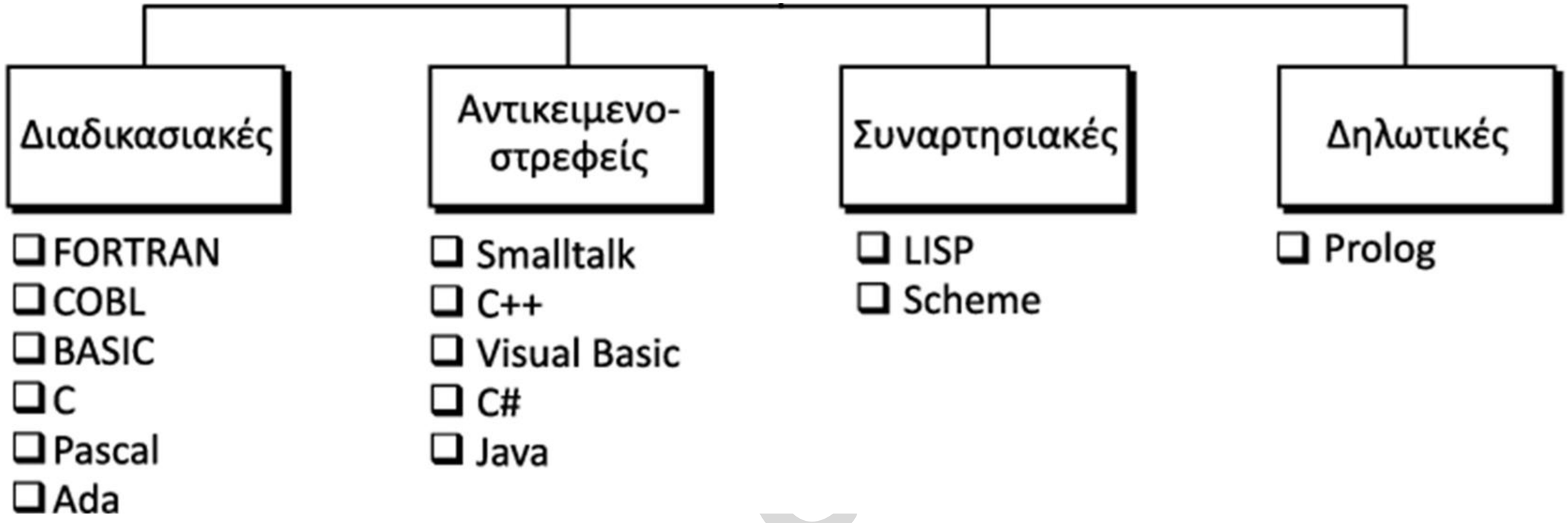
- Παρόλο που οι γλώσσες assembly βελτίωσαν δραστικά την αποδοτικότητα του προγραμματισμού, οι προγραμματιστές έπρεπε ακόμα να δίνουν ιδιαίτερη βαρύτητα στο υλικό που χρησιμοποιούσαν

- Η εργασία με τις συμβολικές γλώσσες ήταν και αυτή πολύ κουραστική, επειδή κάθε εντολή μηχανής έπρεπε να κωδικοποιηθεί ξεχωριστά

9.47

- Για βελτίωση της αποδοτικότητας στον προγραμματισμό και για την αλλαγή της εστίασης του προγραμματιστή στο προς επίλυση πρόβλημα και όχι στον ίδιο τον υπολογιστή, οδήγησαν στην ανάπτυξη των γλωσσών υψηλού επιπέδου (high-level languages), COBOL, Pascal, Ada, C, C++, Java, Python, κ.ά.

...Γλώσσες Προγραμματισμού

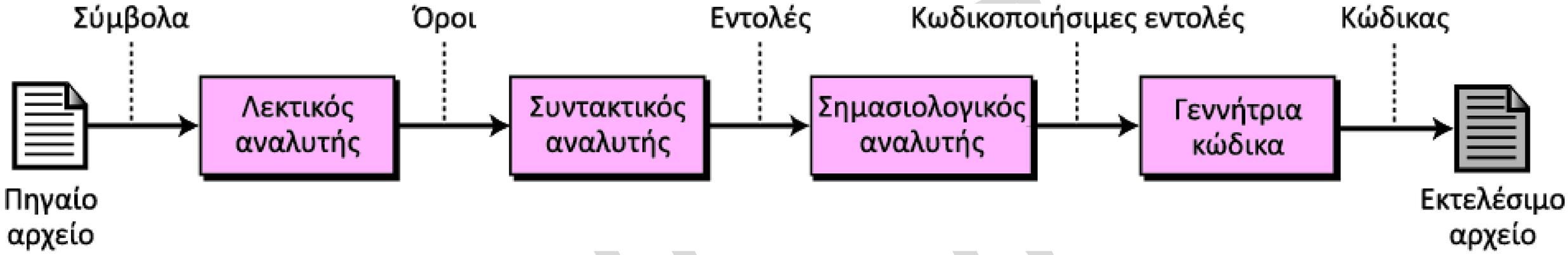


Μεταγλώττιση-Διερμήνευση…

- Τα σύγχρονα προγράμματα συνήθως γράφονται σε κάποια από τις γλώσσες υψηλού επιπέδου
- Για να είναι εφικτή η εκτέλεση του προγράμματος σε έναν υπολογιστή, πρέπει να μεταφραστεί στη γλώσσα μηχανής του υπολογιστή στον οποίο θα εκτελεστεί
- Το πρόγραμμα που βρίσκεται σε μια γλώσσα υψηλού επιπέδου ονομάζεται πηγαίο πρόγραμμα
- Το πρόγραμμα που είναι μεταφρασμένο σε γλώσσα μηχανής ονομάζεται εκτελέσιμο πρόγραμμα ή πρόγραμμα-αντικείμενο (object program)
- Για τη μετάφραση των προγραμμάτων χρησιμοποιούνται δύο μέθοδοι: η μεταγλώττιση (Compilation) και η διερμήνευση (Interpretation).

...Μεταγλώττιση-Διερμήνευση

Μεταγλώττιση



Αλγόριθμοι και Αλγοριθμικές Δομές

Στόχος

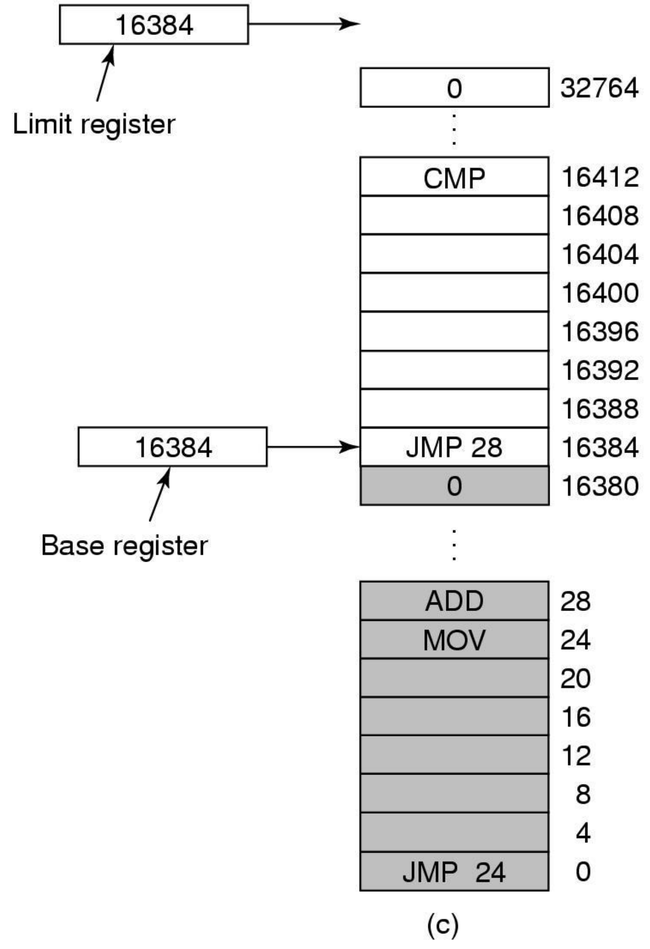
- Να διατυπώσουμε τον ορισμό για τον αλγόριθμο και να τον συσχετίσουμε με τη λύση ενός προβλήματος.
- Να διατυπώσουμε τον ορισμό για καθεμιά από τις τρεις βασικές αλγοριθμικές δομές
  - Ακολουθίας
  - Επιλογής (δομές ελέγχου της ροής)
  - Επανάληψης (Βρόγχοι)

  και να περιγράψουμε τη χρήση τους σε αλγορίθμους.
- Να περιγράψουμε τους τρόπους αναπαράστασης των αλγορίθμων
  - Περιφραστική διατύπωση
  - Διαγράμματα για την αναπαράσταση των αλγορίθμων
  - Ψευδοκώδικας
  - Κωδικοποίηση (συγγραφή προγράμματος σε γλώσσα προγραμματισμού)
- Να γνωρίσουμε βασικούς αλγορίθμους, τον τρόπο σκέψης για την σχεδιάση τους
- Να ανακαλύψουμε τις εφαρμογές τους.
- Να ανακαλύψουμε πως μεταβαίνουμε σε γλώσσα προγραμματισμού

- Αλγόριθμος, είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος
- Κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος είναι:
  - Είσοδος (**Καμιά, μία ή περισσότερες τιμές δεδομένων**)
  - Έξοδο (αποτελέσματα για τον χρήστη ή άλλον αλγόριθμο)
  - Περατότητα (πεπερασμένα βήματα, που εκτελούνται σε πεπερασμένο χρόνο)
  - Καθοριστικότητα (σαφώς καθορισμένες εντολές)
  - Αποτελεσματικότητα (κάθε εντολή καλώς ορισμένη είναι και εκτελέσιμη)
- Περιγραφή & αναπαράσταση αλγορίθμων
  - Ελεύθερο κείμενο (free text): αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου.
  - Διαγραμματικές τεχνικές: συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου (η πιο γνωστή είναι το διάγραμμα ροής- flow chart)
  - Βηματική εκτέλεση σε φυσική γλώσσα (natural language): περιγραφή κατά βήματα.
  - Κωδικοποίηση (coding): ένα πρόγραμμα που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο (σε ψευδογλώσσα ή γλώσσα προγραμματισμού)

- Μεταβλητές
  - Αριθμητικές: που αποθηκεύουν τιμές όπως 70, -32.5, κ.ο.κ.
  - Αλφαριθμητικές: που αποθηκεύουν τιμές, όπως «Γεώργιος», «John», κ.ο.κ.
  - Λογικές με τιμή αληθής ή ψευδής
- Αποτελούν τον φορέα για τη διατήρηση των Δεδομένων και των Αποτελεσμάτων

- Επιλογή ονομάτων μεταβλητών
  - Τα ονόματα των μεταβλητών μπορούν να περιλαμβάνουν πεζά ή κεφαλαία γράμματα, αριθμούς και το χαρακτήρα κάτω παύλα (_) και άλλα σύμβολα
  - Μπορεί να είναι case sensitive (διακρίνει και διαφοροποιεί τα πεζά από τα κεφαλαία, γιατί έχουν διαφορετική ASCII, ανάλογα με τη γλώσσα και το υπολογιστικό σύστημα που υλοποιείται ο αλγόριθμος)
  - Το όνομα πρέπει να ξεκινά από χαρακτήρα και όχι αριθμό χωρίς κενά. Αν επιθυμούμε την ύπαρξη δυο λέξεων τότε χρησιμοποιείται ή κάτω παύλα π.χ. Μέγιστη_Τιμή
  - Είναι προτιμότερο να είναι περιγραφικά και περιγραστικά ονόματα για να μπορούμε να τα θυμόμαστε (με μέτρο στο πλήθος των χαρακτήρων)
  - Δεν επιτρέπεται να χρησιμοποιηθεί ως όνομα μεταβλητής κάποια από τις δεσμευμένες λέξεις της γλώσσας προγραμματισμού ή ακόμη και της ψευδογλώσσας που χρησιμοποιείται για την απεικόνιση του αλγορίθμου
  - Ουσιαστικά, χρησιμοποιούμε μια μεταβλητή για να αποθηκεύσουμε σε αυτήν κάποια τιμή (είσοδο δεδομένων από το χρήστη ή το αποτέλεσμα κάποιας έκφρασης). Όταν χρησιμοποιούμε την μεταβλητή επικαλούμαστε την τιμή που αυτή περιέχει
- Ένας τύπος δεδομένων (data type) καθορίζει ένα σύνολο τιμών και ένα σύνολο λειτουργιών που μπορούν να εφαρμοστούν σε αυτές τις τιμές
- Οι μεταβλητές είναι ονόματα για θέσεις μνήμης



(c)

- Χρησιμοποιείτε όσες μεταβλητές νομίζετε και σας προκύπτει ότι χρειάζεστε!!!

- Άρα προσθέτετε μια κάθε φορά που προκύπτει η ανάγκη, χωρίς φόβο και πάθος και

- Αργότερα και με την αύξηση της εμπειρίας σας (ελπίζουμε σε αυτό το μάθημα και σε αυτό το εξάμηνο!!!) στον προγραμματισμό θα εντοπίζετε πόσες και ποιες πραγματικά χρειάζεστε

- Οι περισσότερες γλώσσες προγραμματισμού επιτρέπουν τον ορισμό σταθερών

- Μια σταθερά, όπως και μια μεταβλητή, έχει όνομα που μπορεί να δεχθεί μια τιμή, η οποία όμως δεν μπορεί να αλλάξει μετά τον ορισμό της στην αρχή του προγράμματος

- Σχεδόν κάθε πρόγραμμα χρειάζεται να διαβάζει και να γράφει δεδομένα. Αυτές οι λειτουργίες μπορεί να γίνουν πολύ σύνθετες, ειδικά όταν διαβάζουμε και γράφουμε μεγάλα αρχεία. Οι περισσότερες γλώσσες προγραμματισμού χρησιμοποιούν προκαθορισμένες συναρτήσεις για την είσοδο και την έξοδο.

- Τα δεδομένα εισάγονται είτε από μια εντολή είτε από μια προκαθορισμένη συνάρτηση, όπως η scanf στη γλώσσα C.

- Τα δεδομένα εξάγονται είτε από κάποια εντολή είτε από μια προκαθορισμένη συνάρτηση, όπως η printf στη γλώσσα C.

- Σημαντικά σημεία για τους αλγόριθμους που θα αναπτύσσετε:
  - Οι εντολές εκτελούνται σειριακά όπως παρατίθενται στον αλγόριθμο/ πρόγραμμα (τουλάχιστον για την ώρα, επειδή αναφερόμαστε σε διαδικασιακό προγραμματισμό)
  - Πρέπει να προβλέπεται κάποιας μορφής εισόδων/ εξόδων
    - Απόδοση τιμών εντός του αλγορίθμου (προγράμματος αργότερα)
    - Πληκτρολόγιο
    - Είσοδος από αρχείο
    - Είσοδος από άλλο τμήμα του προγράμματος
    - Είσοδος από άλλο πρόγραμμα
  - Δεν μπορεί να χρησιμοποιείται ίδιο όνομα για δύο διαφορετικές μεταβλητή
    - Εάν είναι case sensitive μπορεί να γίνει, αλλά θα έχουμε πρόβλημα εμείς
  - Δεσμευμένες λέξεις (εντολές, αλγοριθμικές δομές, κ.λπ.), αυτές οι λέξεις καθώς και όλες όσες είναι έντονα γραμμένες καλούνται δεσμευμένες λέξεις

- Πρόκειται για γνωστά σύμβολα πράξεων (αναλυτικά και στη C θα μιλήσουμε για όλους τους τελεστές σε μερικές εβδομάδες, εδώ εισαγάγουμε όσους χρειαζόμαστε για την αλγοριθμική σκέψη που θα δουλέψουμε σε αυτήν τη φάση)
  - Αριθμητικοί (+, -, *, /, ** για την ύψωση σε δύναμη π.χ. το 8^2 συμβολίζεται με 8**2 (στη C υπάρχουν μερικές διαφοροποιήσεις στον κλασικό τρόπο χρήσης των αριθμητικών- και όχι μόνο τελεστών- αλλά αυτές θα τις αντιμετωπίσουμε αργότερα)
  - Υπάρχουν και οι τελεστές div και mod
    - Ο τελεστής div (διαφορετικές προσεγγίσεις στη C, από / μέχρι div(), div_t(), κ.λπ.) επιστρέφει το **ακέραιο πηλίκο** της διαίρεσης δυο αριθμών
    - Το mod (% στη C) το **ακέραιο υπόλοιπο**

      π.χ. 13 div 3 = 4

      13 mod 3 = 1

      Ο τελεστής mod χρησιμοποιείται ευρέως σε ελέγχους, όπως στις περιπτώσεις όπου πρέπει να διερευνηθεί αν κάποιος αριθμός είναι άρτιος ή περιττός, δηλαδή αν είναι πολλαπλάσιο του 2 ή όχι

      4 mod 2 = 0 ενώ 13 mod 2 = 1

      Παρατηρήσεις για τον τελεστή mod
      - Σε μια πράξη **a mod b** αναμένουμε αποτέλεσμα από 0 μέχρι b-1
      - Αν το a δεν διαιρείται πλήρως με το b τότε παράγει κάποια μη μηδενική ακέραια τιμή
      - Αν το a διαιρείται πλήρως με το b τότε το υπόλοιπο γίνεται 0 (μηδέν)
      - Αν το a είναι κάποιος αριθμός και το b είναι 0, τότε λαμβάνουμε ένα σφάλμα χρόνου μεταγλώττισης (compile-time error)

- Σε περιπτώσεις αρνητικών **a** ή/και **b**, αν και ανάλογα με τη γλώσσα μπορεί να έχουμε διαφοροποιήσεις, θεωρητικά (έτσι υλοποιείται και στη C) αν:
  - ✓ a>0 και b<0 επιστρέφεται τιμή που αντιστοιχεί στο a%(|b|),
  - ✓ a<0 και b>0 επιστρέφεται –((|a|)%b),
  - ✓ a<0 και b<0, με a<b επιστρέφεται –((|a|)%b),
  - ✓ a<0 και b<0, με a>b επιστρέφεται a.

- Συγκριτικοί: >, <, >= , <=, = και <>
- Λογικοί: **ΚΑΙ (AND, &&)**, **Η (OR, ||)**, **ΟΧΙ (NOT, !)** οι οποίοι συνδέουν κυρίως συνθήκες ελέγχου και το αποτέλεσμα είναι πάντα μια λογική τιμή **αληθής** (**true, 1, κ.λπ.**) ή **ψευδής (false, 0, κ.λπ.)**

• Περιπτώσεις Χρήσης DIV και MOD:

- Η απομόνωση των ψηφίων ενός αριθμού απαιτεί συνεχείς διαιρέσεις του αριθμού με τους κατάλληλους αριθμούς (συνήθως 10, 100, 1000, κ.λπ.). Συνήθως ξεκινώντας από τους μεγαλύτερους αριθμούς (διαιρέτες).
- Όταν επιθυμούμε να στρογγυλοποιήσουμε έναν αριθμό.
- Στο έλεγχο εάν ένας αριθμός είναι πολλαπλάσιος ή υποδιαίρεση κάποιου άλλου, οπότε το υπόλοιπο μιας ακέραιας διαίρεσης των δύο αριθμών θα πρέπει να είναι μηδέν. Τέτοιες περιπτώσεις είναι και ο χωρισμός σε συγκεκριμένες κατηγορίες συσκευασιών.
- Στον έλεγχο άρτιων και περιττών.

Τελεστής	*Ορισμός*	*Παράδειγμα*
+	Πρόσθεση	3 + 5
–	Αφαίρεση	2 – 4
*	Πολλαπλασιασμός	Num * 5
/	Διαίρεση (το αποτέλεσμα είναι το πηλίκο)	Sum / Count
%	Διαίρεση (το αποτέλεσμα είναι το υπόλοιπο)	Count % 4
++	Αύξηση κατά ένα (πρόσθεση του 1 στην τιμή της μεταβλητής)	Count++
– –	Μείωση κατά ένα (αφαίρεση του 1 από την τιμή της μεταβλητής)	Count– –

Τελεστής	*Ορισμός*	*Παράδειγμα*
<	Μικρότερο από	Num1 < 5
<=	Μικρότερο από ή ίσο με	Num1 <= 5
>	Μεγαλύτερο από	Num2 > 3
>=	Μεγαλύτερο από ή ίσο με	Num2 >= 3
==	Ίσο με	Num1 == Num2
!=	Διάφορο του	Num1 != Num2

...Τελεστές

Λογικοί

Τελεστής	*Ορισμός*	*Παράδειγμα*
!	Not	! (Num1 < Num2)
&&	And	(Num1 < 5) && (Num2 > 10)
||	Or	(Num1 < 5) || (Num2 > 10)

- Ιεραρχία Τελεστών
  1. δύναμη
  2. πολλαπλασιασμός, διαίρεση, div, mod
  3. πρόσθεση και αφαίρεση
  4. συνθήκες (τελεστές σύγκρισης)
  5. λογικές πράξεις

- ΠΡΟΣΟΧΗ:!!! Η παρένθεση αλλάζει την ιεραρχία αυτή.

- Σε ότι αφορά την ιεραρχία των κατηγοριών των πράξεων, αυτή είναι:
  - Πρώτα εκτελούνται οι αριθμητικοί τελεστές
  - Έπειτα οι συγκριτικοί και τέλος
  - Οι Λογικοί

Παραστάσεις-Εκφράσεις

- Συνδυασμοί μεταβλητών ή σταθερών και τελεστών
- Το αποτέλεσμα μιας έκφρασης αποδίδεται σε μια μεταβλητή με εκχώρηση (ή ανάθεση) τιμής

π.χ. τιμή = α + β,

- όπου το αποτέλεσμα του αθροίσματος των τιμών των μεταβλητών α και β θα εκχωρηθεί (ανατεθεί) στη μεταβλητή τιμή
- Η τελική τιμή μια εκχώρησης εξαρτάται από την ιεραρχία των πράξεων και τις παρενθέσεις
- Αποδεκτές εκφράσεις

τιμή = 5

τιμή = «αρκετά»

τιμή = α * β

τιμή = τιμή + 3

- Μη αποδεκτές εκφράσεις

τιμή = 5 + «χ»

α + τιμή = 6

Να διαβασθούν δύο αριθμοί, να υπολογισθεί και να εκτυπωθεί το άθροισμά τους

```
Αλγόριθμος Άθροισμα
   Διάβασε a
   Διάβασε b
   c = a + b
   Εκτύπωσε c
Τέλος Άθροισμα
```

Ευχαριστώ

Ερωτήσεις?