

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΠΛΗΡΟΦΟΡΙΚΗ

(Τάξη: Γ΄)

ΓΕΝΙΚΟ
ΛΥΚΕΙΟ

2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

ΕΙΔΙΚΟΙ ΕΠΙΣΤΗΜΟΝΕΣ

ΑΞΙΟΛΟΓΗΣΗΣ:

Μαυρίδης Ιωάννης, Av. Καθηγητής Πανεπιστημίου Μακεδονίας (Συντονιστής)

Βαρζάκας Παναγιώτης, Av. Καθηγητής ΑΤΕΙ Στερεάς Ελλάδας

Μαραγκός Κωνσταντίνος, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ19

Παπαδάκης Σπυρίδων, Σχολικός Σύμβουλος ΠΕ19

ΕΠΙΣΤΗΜΟΝΙΚΗΣ
ΕΠΟΠΤΕΙΑΣ:

Μπελεσιώτης Βασίλειος, Σχολικός Σύμβουλος ΠΕ19

ΕΚΠΟΝΗΣΗΣ:

Κανίδης Ευάγγελος, Σχολικός Σύμβουλος ΠΕ19

Λέκκα Δήμητρα, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ19

Τζήμας Δημήτριος, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ19

**«ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα) – Νέο Πρόγραμμα Σπουδών»
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ «ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ»**

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ
Σωτήριος Γκλαβάς
Πρόεδρος του Ινστιτούτου Εκπαιδευτικής Πολιτικής

Υπεύθυνη Πράξης
Γεωργία Φέρμελη
Σύμβουλος Α΄ Ινστιτούτου Εκπαιδευτικής Πολιτικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Το παρόν συγχρηματοδοτήθηκε από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και εθνικούς πόρους στο πλαίσιο της πράξης «**ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα) – Νέο Πρόγραμμα Σπουδών**» του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Διά Βίου Μάθηση**»

**Πληροφορική
Γ΄ τάξης
Ομάδας Προσανατολισμού των Θετικών
Σπουδών Γενικού Λυκείου**

Πρόγραμμα Σπουδών του μαθήματος «Πληροφορική» Γ΄ τάξης ΓΕΛ

Το παρόν Πρόγραμμα Σπουδών (Π.Σ.) αφορά το μάθημα «Πληροφορική» της Γ΄ τάξης ΓΕΛ που ορίστηκε με το Νόμο 4310 (ΦΕΚ 258/8-12-2014).

Έχει σκοπό οι μαθητές να αναπτύξουν αναλυτική και συνθετική σκέψη, να αποκτήσουν ικανότητες μεθοδολογικού χαρακτήρα και να μπορούν να επιλύουν προβλήματα και να δημιουργούν τα αντίστοιχα προγράμματα σε προγραμματιστικό περιβάλλον. Παρέχει ένα επιστημονικό υπόβαθρο για την Επιστήμη Υπολογιστών/Πληροφορική και την αξιοποίησή της σε άλλες επιστήμες, παράλληλα με μια εφαρμοσμένη προσέγγιση όπου χρησιμοποιείται μια πραγματική γλώσσα προγραμματισμού.

Συνοδεύεται από τον Οδηγό Εκπαιδευτικού, ο οποίος στοχεύει στην ανάδειξη της γενικής φιλοσοφίας και του εκπαιδευτικού προσανατολισμού του Π.Σ., καθώς και των διδακτικών προσεγγίσεων που προτείνονται. Επίσης, στοχεύει στην αποτελεσματική υποστήριξη των εκπαιδευτικών κατά το σχεδιασμό και την οργάνωση, τόσο της διδασκαλίας τους όσο και της μαθησιακής πορείας και καθοδήγησης των μαθητών, σύμφωνα με τους στόχους και τις παιδαγωγικές στρατηγικές του νέου Π.Σ.

ΣΤΟΧΟΙ Οι μαθητές πρέπει να είναι ικανοί:	ΘΕΜΑΤΙΚΕΣ ΕΝΟΤΗΤΕΣ [Ενδεικτικές Ώρες (εύρος) προόδου ύλης + Ενδεικτικές Ώρες (εύρος) εμπέδωσης ύλης]	ΠΑΡΑΤΗΡΗΣΕΙΣ/ΕΝΔΕΙΚΤΙΚΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ
<ul style="list-style-type: none">• Να περιγράφουν τι περιλαμβάνει η έννοια της υπολογιστικής σκέψης (ΥΣ).• Να περιγράφουν τη σημασία της ΥΣ στην επίλυση προβλημάτων της καθημερινής ζωής.• Να περιγράφουν τα χαρακτηριστικά της υπολογιστικής σκέψης.• Να αναλύουν ένα υπολογιστικό πρόβλημα περιγράφοντας τις βασικές διαδικασίες με τις οποίες αντιμετωπίζεται (διάσπαση, αναγνώριση προτύπων, γενίκευση και σχεδίαση αλγορίθμου).	1. Εισαγωγή στην Υπολογιστική Σκέψη [4 (από 3 έως 5) + 0 Ω] 1.1 Η έννοια της υπολογιστικής σκέψης 1.2 Χαρακτηριστικά (π.χ. αφαίρεση - abstraction, μοντελοποίηση, γενίκευση, μετασχηματισμός, πρότυπα - patterns). 1.3 Υπολογιστική σκέψη και επίλυση προβλημάτων	<ul style="list-style-type: none">• Επεξήγηση σκοπού και στόχων μαθήματος και τρόπου εργασίας.• Σύνδεση υπολογιστικής σκέψης με την επίλυση προβλημάτων καθημερινότητας και από τη σχολική εμπειρία των μαθητών μέσω συζήτησης και καταιγισμού ιδεών.• Μέσα από μια μελέτη περίπτωσης ενός σύνθετου προβλήματος από τις εμπειρίες και τις γνώσεις των μαθητών, οι μαθητές, εργαζόμενοι σε ομάδες, αναλύουν το πρόβλημα σε απλούστερα, εντοπίζουν τα δεδομένα και σχεδιάζουν την επεξεργασία που απαιτείται για την επίλυση του προβλήματος.

<ul style="list-style-type: none"> • Να περιγράφουν με τη βοήθεια ενός διαστρωματικού αφαιρετικού μοντέλου μελέτης τα βασικά στοιχεία και τις λειτουργίες που απαιτούνται για την επίλυση ενός προβλήματος μέσω υπολογιστή. • Να εξηγούν την οργάνωση της κλασικής von Neumann μηχανής και των κύριων λειτουργικών μονάδων της. • Να επεξηγούν το ρόλο και τους περιορισμούς του υλικού και του λογισμικού στην επίλυση προβλημάτων σε ένα υπολογιστικό σύστημα. • Να περιγράφουν τον τρόπο που τα δεδομένα (ακέραιοι, πραγματικοί, χαρακτήρες, κλπ) αναπαριστώνται σε έναν υπολογιστή. • Να αντιστοιχίζουν τους τύπους δεδομένων με ανάλογες εντολές ISA. 	<p>2. Υπολογιστής Γενικού Σκοπού: Υλικό και Λογισμικό [6 (5-7) + 4 (3-5) Ω]</p> <p>2.1 Από την επιθυμητή συμπεριφορά (λύση προβλήματος) στα ηλεκτρονικά στοιχεία</p> <p>2.2 Επίπεδα αφαίρεσης:</p> <p>2.2.1 Πρόβλημα 2.2.2 Αλγόριθμος 2.2.3 Πρόγραμμα 2.2.4 Αρχιτεκτονική Μηχανής (ISA) 2.2.5 Μικρο-αρχιτεκτονική 2.2.6 Λογικά Κυκλώματα 2.2.7 Ηλεκτρονικά Στοιχεία</p> <p>2.3 Μοντέλο von Neumann</p> <p>2.4 Αντιστοιχία τύπων δεδομένων και εντολών ISA (αναφορά)</p> <p>2.5 Αναπαράσταση δεδομένων</p> <p>2.5.1 Δυαδικό, δεκαδικό 2.5.2 Ακέραιοι αριθμοί</p> <ul style="list-style-type: none"> • μη προσημασμένοι • συμπλήρωμα ως προς 1 και 2 • αριθμητικές και λογικές πράξεις • υπερχείλιση (overflow) <p>2.5.3 Πραγματικοί αριθμοί (κινητής υποδιαστολής με μονή καθώς και με διπλή ακρίβεια) 2.5.4 Οκταδικό, δεκαεξαδικό 2.5.5 Unicode, ASCII</p>	<ul style="list-style-type: none"> • Οι μαθητές εργαζόμενοι σε ομάδες να αναζητήσουν πηγές και πόρους για α) την αρχιτεκτονική von Neumann και β) τα επίπεδα αφαιρετικής μελέτης ενός υπολογιστικού συστήματος. • Επίδειξη του κύκλου εκτέλεσης εντολής (π.χ $c = a + b$). • Με ενδεικτικά παραδείγματα να εξηγηθούν οι τρόποι αναπαράστασης των δεδομένων. • Με χρήση Hexdump να διερευνήσουν την αναπαράσταση του περιεχομένου διαφόρων τύπων αρχείων.
<ul style="list-style-type: none"> • Να περιγράφουν τους βασικούς τύπους δεδομένων και τις βασικές εντολές της γλώσσας προγραμματισμού. • Να αναγνωρίζουν και να χρησιμοποιούν τις βασικές μαθηματικές συναρτήσεις γλώσσας προγραμματισμού. • Να χρησιμοποιούν το διερμηνευτή γλώσσας προγραμματισμού για να κάνουν απλές πράξεις. • Να χρησιμοποιούν το προγραμματιστικό περιβάλλον για σύνταξη κώδικα προγράμματος και μετέπειτα εκτέλεσής του. 	<p>3. Βασικά στοιχεία γλώσσας προγραμματισμού [2 (1-3) + 4 (3-5) Ω]</p> <p>3.1 Τύποι δεδομένων</p> <p>3.2 Βασικές εντολές</p> <p>3.3 Αριθμητικές και λογικές πράξεις</p> <p>3.4 Βασικές συναρτήσεις (έτοιμες μαθηματικές)</p> <p>3.5 Περιβάλλοντα προγραμματισμού</p> <p>3.6 Απλά προγράμματα</p>	<ul style="list-style-type: none"> • Οι μαθητές εργαζόμενοι ατομικά ή ανά δύο (pair programming) να πειραματιστούν με τον διερμηνευτή της γλώσσας εκτελώντας πράξεις με χρήση των δομικών στοιχείων της Python (η Python ως αριθμομηχανή). • Οι μαθητές, με χρήση του περιβάλλοντος προγραμματισμού της Python IDLE, αναπτύσσουν απλά προγράμματα (αποθηκευμένα σενάρια εντολών), ως μια πρώτη πρακτική άσκηση για την επίλυση προβλημάτων.
<ul style="list-style-type: none"> • Να περιγράφουν τον τρόπο αναπαράστασης της μεταβλητής στην κύρια μνήμη. • Να περιγράφουν τον τρόπο αποθήκευσης τιμών στην κύρια μνήμη ώστε να υλοποιούν τη δομή του Πίνακα και της Λίστας. • Να αναγνωρίζουν τους διαφορετικούς τύπους αρχείων και να τους αντιστοιχούν με τις 	<p>4. Λογική οργάνωση δεδομένων και αποθήκευσή τους [4 (3-5) + 4 (3-5) Ω]</p> <p>4.1 Κύρια Μνήμη</p> <p>4.1.1 Μεταβλητές 4.1.2 Δομές Δεδομένων:</p> <ul style="list-style-type: none"> • Στατικές: πίνακας • Δυναμικές: λίστα <p>4.2 Αρχεία</p>	<ul style="list-style-type: none"> • Να επιδειχθούν (π.χ. με προσομοίωση) α) η δομή της κύριας μνήμης και διευθύνσεις και β) οι διαφορές αποθήκευσης μεταξύ πίνακα και λίστας. • Να γίνει επίδειξη προσομοίωσης της λειτουργίας ενός αλγορίθμου συμπίεσης. • Να αναζητήσουν-πειραματιστούν οι μαθητές, εργαζόμενοι σε ομάδες,

<p>εφαρμογές που τα δημιούργησαν.</p> <ul style="list-style-type: none"> • Να αναφέρουν και επεξηγούν τρόπους και μεθοδολογίες κωδικοποίησης και συμπίεσης δεδομένων σε αρχεία. 	<p>4.2.1 Τύποι αρχείων και αντιστοιχία με εφαρμογές. 4.2.2 Συμπίεση</p> <ul style="list-style-type: none"> • μη-απωλεστική • απωλεστική 	<p>αλγόριθμους απωλεστικής (π.χ. αρχεία MP3) και μη απωλεστικής (LZW, RLE) συμπίεσης και να τις συνδέσουν με μορφές αρχείων που χρησιμοποιούν συχνά (εικόνας, βίντεο).</p>
<ul style="list-style-type: none"> • Να αναγνωρίζουν ένα σύνθετο πρόβλημα και τα πλεονεκτήματα που προκύπτουν από την ανάλυσή του σε απλούστερα. • Να επιλύουν προβλήματα χρησιμοποιώντας τα βήματα ανάλυσης σε επιμέρους προβλήματα, με αξιοποίηση υπαρχουσών λύσεων. • Να επιλέγουν κατά τον σχεδιασμό αλγορίθμου τις κατάλληλες αλγοριθμικές δομές, όπως η ακολουθία, η επιλογή, η επανάληψη και η αναδρομή. • Να περιγράφουν έναν αλγόριθμο μέσω ψευδοκώδικα ή διαγράμματος ροής. 	<p>5. Από το πρόβλημα στον αλγόριθμο [8 (7-9)+4 (3-5) Ω]</p> <p>5.1 Εισαγωγή στη διαχείριση της πολυπλοκότητας ενός προβλήματος 5.1.1 Η αξία της αφαίρεσης 5.1.2 Ανάλυση ενός προβλήματος σε απλούστερα υποπροβλήματα</p> <p>5.2 Περιγραφή με ψευδοκώδικα / διάγραμμα ροής</p> <p>5.3 Αλγοριθμικές δομές: ακολουθία, επιλογή, επανάληψη, αναδρομή</p>	<ul style="list-style-type: none"> • Να συζητηθούν παραδείγματα απλών και πολύπλοκων - σύνθετων προβλημάτων. • Μέσω παραδείγματος να αναδειχθεί η αξία της «αφαίρεσης» για τη διαχείριση της πολυπλοκότητας ενός προβλήματος. • Επίδειξη έτοιμων αλγορίθμων σε ψευδοκώδικα και διάγραμμα ροής. • Εξάσκηση των μαθητών στην αξιοποίηση βασικών αλγοριθμικών δομών, με εναλλακτικούς τρόπους παρουσίασης.
<ul style="list-style-type: none"> • Να επεξηγούν τη λογική συγγραφής προγραμμάτων ανά είδος προγραμματισμού (Διαδικαστικός, Αντικειμενοστρεφής και Συναρτησιακός προγραμματισμός). • Να υλοποιούν έναν πλήρη κύκλο ανάπτυξης προγράμματος σε περιβάλλον προγραμματισμού (συγγραφή κώδικα, αποσφαλμάτωση, μετάφραση/διερμηνεία, εκτέλεση). • Να επιλέγουν και να χρησιμοποιούν βασικές αλγοριθμικές δομές, όπως ακολουθία, επιλογή, επανάληψη και δομές δεδομένων, όπως οι Συμβολοσειρές και οι Λίστες στη γλώσσα προγραμματισμού. 	<p>6. Από τον αλγόριθμο στο πρόγραμμα [6 (5-7) + 8 (7-9) Ω]</p> <p>6.1 Η λογική συγγραφής προγράμματος ανάλογα με το είδος προγραμματισμού (η χρήση της γλώσσας προγραμματισμού ως εργαλείο έκφρασης της σχεδίασης)</p> <ul style="list-style-type: none"> • Διαδικαστικός προγραμματισμός • Αντικειμενοστρεφής προγραμματισμός • Συναρτησιακός προγραμματισμός <p>6.2 Διαδικασία συγγραφής, μετάφρασης και εκτέλεσης προγράμματος</p> <p>6.3 Σύνταξη εντολών γλώσσας προγραμματισμού 6.3.1 Αλγοριθμικές δομές (ακολουθία, επιλογή, επανάληψη) 6.3.2 Δομές δεδομένων</p> <ul style="list-style-type: none"> • Συμβολοσειρές (ή Αλφαριθμητικά) (Strings) • Λίστες (Επεξεργασία λιστών, Τελεστές, Μέθοδοι) 	<ul style="list-style-type: none"> • Αναζήτηση πληροφοριών (ιστοεξερεύνηση) για τα είδη προγραμματισμού, όπως ο Διαδικαστικός, Αντικειμενοστρεφής, Συναρτησιακός προγραμματισμός και του τρόπου λειτουργίας του καθενός. • Επίδειξη σε Python και εφαρμογή της διαδικασίας συγγραφής, μετάφρασης και εκτέλεσης προγράμματος. • Παρουσίαση και πειραματισμός των αλγοριθμικών δομών της Python με παράλληλη αναφορά σε αντίστοιχες δομές της ψευδογλώσσας. • Ανάλυση των βασικών κανόνων συγγραφής ενός προγράμματος σε Python (χρήση εσοχών, σχόλια, εισαγωγικά κ.λπ). • Επεξεργασία συμβολοσειρών (π.χ συνένωση). • Βασικές λειτουργίες σε λίστες (εμφάνιση, εισαγωγή, διαγραφή στοιχείου κ.λπ).
<ul style="list-style-type: none"> • Να αναγνωρίζουν αντικείμενα και να περιγράφουν τα χαρακτηριστικά τους. • Να αναγνωρίζουν τις συναρτήσεις και τη χρήση των παραμέτρων. • Να συντάσσουν συναρτήσεις που πραγματοποιούν μια συγκεκριμένη εργασία και να αξιοποιούν τα πλεονεκτήματα που προκύπτουν. • Να εντοπίζουν-τεκμηριώνουν 	<p>7. Προηγμένα στοιχεία γλώσσας προγραμματισμού [6 (5-7) + 8 (7-9) Ω]</p> <p>7.1 Αντικείμενα 7.2 Ορισμός συναρτήσεων</p> <ul style="list-style-type: none"> • τυπικές-πραγματικές παράμετροι • εμβέλεια μεταβλητών <p>7.3 Αναδρομή 7.4 Δομημένη συγγραφή- αρθρώματα</p>	<ul style="list-style-type: none"> • Αναφορά στις έννοιες Αντικείμενο και Μέθοδοι του αντικειμένου, με παραδείγματα. • Να επεξηγηθεί ότι όλα στην Python είναι αντικείμενα με μεθόδους επί αυτών. • Παρουσίαση παραδειγμάτων με αντικείμενα (συμβολοσειρές, λίστες κ.λπ). • Εξάσκηση στο χειρισμό μεθόδων επί αντικειμένου (π.χ μέθοδοι χειρισμού

<p>την εμβέλεια των μεταβλητών σε ένα πρόγραμμα ή σε μια συνάρτηση.</p> <ul style="list-style-type: none"> • Να εξηγούν απλούς αναδρομικούς αλγορίθμους. • Να εφαρμόζουν λειτουργίες στις δομές δεδομένων: Λεξικό, Πλειάδα. • Να εφαρμόζουν βασικές λειτουργίες σε αρχεία που χρησιμοποιεί η γλώσσα προγραμματισμού. 	<p>(modules)</p> <p>7.5 Ενσωματωμένες (Built-in) δομές δεδομένων</p> <ul style="list-style-type: none"> • Λεξικό • Πλειάδες (Tuples) <p>7.6 Αρχεία δεδομένων</p> <ul style="list-style-type: none"> • Τύποι αρχείων στη γλώσσα προγραμματισμού • Χειρισμός αρχείων (άνοιγμα, κλείσιμο) • Λειτουργίες σε αρχεία (ανάγνωση - εγγραφή κλπ.) 	<p>συμβολοσειράς).</p> <ul style="list-style-type: none"> • Να εξηγηθεί ο ορισμός συναρτήσεων, η χρήση των παραμέτρων, η κλήση τους και η επιστροφή τιμών καθώς και η εμβέλεια των μεταβλητών. • Επίδειξη αναδρομικών αλγορίθμων και υλοποίηση με παράδειγμα (π.χ. αριθμοί Fibonacci). • Επίδειξη χρήσης βιβλιοθηκών με παραδείγματα. • Εξάσκηση σε χειρισμό βασικών λειτουργιών σε δομές δεδομένων όπως το Λεξικό και η Πλειάδα. • Αναφορά σε τύπους αρχείων στην Python, δυνατούς τρόπους χειρισμού τους (άνοιγμα, κλείσιμο) και λειτουργίες (ανάγνωση, εγγραφή). • Εξάσκηση στο χειρισμό αρχείων.
<ul style="list-style-type: none"> • Να περιγράφουν τους τύπους δομών δεδομένων: Πίνακας, Σύνολο, Στοιβά - Σωρός - Ουρά, Δένδρο, Γράφος. • Να συγκρίνουν και να επιλέγουν την κατάλληλότερη δομή ανάλογα με τον τύπο του προβλήματος. • Να εφαρμόζουν βασικές λειτουργίες σε δομές δεδομένων. 	<p>8. Δομές δεδομένων [10 (9-11) + 4 (3-5) Ω]</p> <p>8.1 Πίνακες (μονοδιάστατοι - πολυδιάστατοι)</p> <p>8.2 Σύνολα</p> <p>8.3 Στοιβά - Σωρός - Ουρά</p> <p>8.4 Δένδρα</p> <p>8.5 Γράφοι</p> <ul style="list-style-type: none"> ▪ Πίνακας Γειτνίασης ▪ Λίστα Γειτνίασης ▪ Σύγκριση αναπαραστάσεων (μνήμη και πολυπλοκότητα βασικών λειτουργιών) 	<ul style="list-style-type: none"> • Μέσω παραδειγμάτων σχεδιασμού σε ψευδογλώσσα και υλοποίησης σε Python να αναπαρασταθούν δομές δεδομένων όπως: πίνακας, σύνολο, στοιβά, ουρά, δένδρο, γράφος (π.χ. δίκτυο δεδομένων ή οδικό δίκτυο με γράφους ή ιεραρχική δομή ενός συστήματος αρχείων). • Μέσω παραδειγμάτων να συγκρίνουν και να επιλέγουν (ανάπτυξη προβληματισμού) την κατάλληλότερη δομή ανάλογα με τον τύπο του προβλήματος. • Να εφαρμόσουν ομαδοσυνεργατικά βασικές λειτουργίες στις δομές δεδομένων (είσοδος, εύρεση κλπ).
<ul style="list-style-type: none"> • Να υλοποιούν σε γλώσσα προγραμματισμού τους κλασικούς αλγορίθμους: Αναζήτησης, Ταξινόμησης και Διάσχισης Γράφων. • Να συγκρίνουν και να επιλέγουν τον κατάλληλο αλγόριθμο ανάλογα με το είδος του προβλήματος. 	<p>9. Κλασικοί Αλγόριθμοι [12 (10-14)+6 (5-7) Ω]</p> <p>9.1 Αναζήτηση</p> <ul style="list-style-type: none"> ▪ σειριακή ▪ δυαδική <p>9.2 Ταξινόμηση</p> <ul style="list-style-type: none"> ▪ γρήγορη (quick) ▪ με συγχώνευση (merge) <p>9.3 Διάσχιση γράφων</p> <ul style="list-style-type: none"> ▪ κατά πλάτος ▪ κατά βάθος 	<ul style="list-style-type: none"> • Δραστηριότητα (ενοιολογική χαρτογράφηση) αναζήτησης και κατηγοριοποίησης κλασικών αλγορίθμων, με τους μαθητές να εργάζονται σε ομάδες. • Δραστηριότητα σχεδιασμού κλασικών αλγορίθμων σε Ψευδογλώσσα και υλοποίηση σε Python με ή χωρίς αναδρομικότητα.
<ul style="list-style-type: none"> • Να συγκρίνουν την επίδοση δύο αλγορίθμων για την επίλυση συγκεκριμένου προβλήματος (benchmarking). • Να αναφέρουν περιορισμούς ως προς το χώρο και το χρόνο ανά κατηγορία αλγορίθμων. • Να υπολογίζουν την επίδοση αλγορίθμων αναζήτησης και ταξινόμησης. • Να εξηγούν τι σημαίνει ανάλυση 	<p>10. Επίδοση και πολυπλοκότητα Αλγορίθμων [12 (10-14)+8 (7-9) Ω]</p> <p>10.1 Περιορισμοί ως προς χρόνο και χώρο</p> <p>10.2 Ανάλυση πολυπλοκότητας ως προς χρόνο/χώρο</p> <p>10.2.1 Υπολογισμός βημάτων:</p> <ul style="list-style-type: none"> ▪ αναζήτηση <ul style="list-style-type: none"> ο σειριακή (sequential search) 	<ul style="list-style-type: none"> • Να δοθούν- συζητηθούν παραδείγματα που αναδεικνύουν την σχέση χώρου-χρόνου στην εκτέλεση αλγορίθμων. • Πρακτικό πρόβλημα, π.χ. password cracking με αργή κρυπτογράφηση (25 φορές DES) σε Python (βήματα: δημιουργία υποψήφιου συνθηματικού, κρυπτογράφηση, σύγκριση με αποθηκευμένο) <ul style="list-style-type: none"> ο εξαντλητική αναζήτηση (exhaustive search) - online

<p>χειρότερης περίπτωσης ενός αλγορίθμου.</p>	<ul style="list-style-type: none"> ο δυαδική (binary search) ▪ ταξινόμηση <ul style="list-style-type: none"> ο γρήγορη ταξινόμηση (quick sort) ο ταξινόμηση με συγχώνευση (merge sort) <p>10.2.2 Ανάλυση χειρότερης περίπτωσης 10.2.3 Ασυμπτωτικός συμβολισμός 10.3.4 Βασικές ασυμπτωτικές κλάσεις αποδοτικότητας</p>	<p>(χρόνος)</p> <ul style="list-style-type: none"> ο αναζήτηση με προϋπολογισμένους πίνακες (dictionary) - offline (χώρος) ο Αξιολόγηση/Σύγκριση ● Μελέτη περίπτωσης-Υπολογισμός βημάτων για την ανάλυση αλγορίθμων αναζήτησης και ταξινόμησης ως προς το χρόνο εκτέλεσης. ● Μελέτη επαναληπτικού και αναδρομικού αλγορίθμου της δυαδικής αναζήτησης ● Παρουσίαση και συζήτηση για κλασικά προβλήματα και υπολογισμός πολυπλοκότητας αντίστοιχων αλγορίθμων (πχ γρήγορη ταξινόμηση, αλγόριθμος του Ευκλείδη κλπ).
<ul style="list-style-type: none"> ● Να περιγράφουν τις διαφορές ανάμεσα στον Διαδικαστικό και τον Αντικειμενοστρεφή προγραμματισμό. ● Να αξιοποιούν τεχνικές του αντικειμενοστρεφούς προγραμματισμού. ● Να ορίζουν και να χρησιμοποιούν κλάσεις καθώς και ιδιότητες και μεθόδους. ● Να ξεχωρίζουν την έννοια της κλάσης από εκείνη του αντικειμένου. ● Να αντιλαμβάνονται το ρόλο της κληρονομικότητας και του πολυμορφισμού στη δημιουργία επαναχρησιμοποιήσιμου κώδικα. 	<p>11. Αντικειμενοστρεφής Προγραμματισμός [10 (9-11) + 6 (5-7) Ω]</p> <p>11.1 Αντικείμενα και κλάσεις 11.2 Στιγμιότητα (αυτόματη αρχικοποίηση αντικειμένων) 11.3 Ιδιότητες και μέθοδοι 11.4 Κληρονομικότητα και πολυμορφισμός 11.5 Ενθυλάκωση και απόκρυψη πληροφορίας</p>	<ul style="list-style-type: none"> ● Μέσω ενδεικτικού παραδείγματος, να αναδειχθούν οι έννοιες του αντικειμενοστρεφούς προγραμματισμού π.χ η κλάση Όχημα (Αυτοκίνητα, αεροπλάνα, πλοία) ή η κλάση Ζώο (Θηλαστικά, πουλιά κ.λπ). ● Να εξηγηθεί, με τη χρήση παραδειγμάτων σε Python, ο τρόπος ορισμού κλάσεων, ιδιοτήτων και μεθόδων και ως “μαύρα κουτιά” να εισαχθεί η έννοια των υποκλάσεων και της κληρονομικότητας. ● Οι μαθητές να εξασκηθούν στη δημιουργία κλάσεων και στη χρήση αντικειμένων μέσα από παραδείγματα από την καθημερινή ζωή που είναι κοντά στα βιώματα τους.
<ul style="list-style-type: none"> ● Να διακρίνουν τα πλεονεκτήματα χρήσης ολοκληρωμένου περιβάλλοντος ανάπτυξης. ● Να περιγράφουν τα εργαλεία που περιλαμβάνει ένα ολοκληρωμένο περιβάλλον ανάπτυξης. ● Να συγγράφουν ποιοτικό κώδικα ακολουθώντας κανόνες ορθής συγγραφής (ονόματα μεταβλητών, εσοχές, τεκμηρίωση). ● Να εφαρμόζουν τεχνικές αποσφαλμάτωσης προγραμμάτων με χρήση εργαλείων ενός περιβάλλοντος ανάπτυξης. 	<p>12. Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης λογισμικού – IDE [4 (3-5) + 2 (1-3) Ω]</p> <p>12.1 Περιβάλλοντα - Εργαλεία 12.2 Ορθή συγγραφή κώδικα 12.2 Τεχνικές Αποσφαλμάτωσης</p>	<ul style="list-style-type: none"> ● Αναζήτηση και οργάνωση πληροφοριών για την έννοια του Ολοκληρωμένου Περιβάλλοντος Ανάπτυξης (Integrated Development Environment- IDE). Συμπλήρωση πίνακα σχετικά με το ρόλο, τη χρησιμότητά τους και τα μέρη που αποτελούνται. ● Σύναψη συμβολαίου κανόνων συγγραφής ποιοτικού κώδικα λαμβάνοντας υπόψη διεθνή πρότυπα. ● Σύγκριση, ως προς τις δυνατότητες, διαφορετικών προγραμματιστικών περιβαλλόντων για τη γλώσσα Python. ● Να δοθούν παραδείγματα προγραμμάτων στα οποία να εντοπιστούν και να συζητηθούν περιπτώσεις συντακτικών και λογικών λαθών.

<ul style="list-style-type: none"> • Να χρησιμοποιούν περιβάλλοντα, Application Program Interfaces (APIs) και βιβλιοθήκες για την ανάπτυξη και τροποποίηση εφαρμογών λογισμικού. • Να ενισχύσουν την δημιουργική τους σκέψη, να ελέγξουν μεγάλο όγκο πληροφοριών και τις διασυνδέσεις τους μέσω της οπτικοποίησης δεδομένων • Να αναγνωρίσουν την συμβολή της πληροφορικής στην επίλυση μεγάλων επιστημονικών διαθεματικών προβλημάτων μέσω της οπτικοποίησης δεδομένων. • Να αποκτήσουν εμπειρία στην συγγραφή και διαχείριση ενός σύνθετου, διαθεματικού έργου. 	<p>13. Εφαρμογές σε Γλώσσα Προγραμματισμού με χρήση API [2 (1-3) + 6 (5-7) Ω]</p>	<ul style="list-style-type: none"> • Παρουσίαση - επεξήγηση της Διεπαφής Προγραμματισμού Εφαρμογών (Application Programming Interface - API). • Αναζήτηση έτοιμων APIs υπηρεσιών κοινωνικής δικτύωσης τα οποία επιτρέπουν την ανάπτυξη εφαρμογών ανάσχυσης από αυτές στοιχείων. • Ανάπτυξη εφαρμογής όπως: <ul style="list-style-type: none"> • Εφαρμογή οπτικοποίησης δεδομένων. • Γραφική Διεπαφή Χρήστη(GUI). • Διαδικτυακή εφαρμογή.
<ul style="list-style-type: none"> • Να περιγράφουν τι είναι ένα Πληροφοριακό Σύστημα (ΠΣ). • Να απαριθμούν τα συστατικά στοιχεία ενός ΠΣ. • Να γνωρίσουν τις κατηγορίες εφαρμογής ενός ΠΣ και τις περιοχές που αξιοποιείται. • Να κατονομάζουν και να αναλύουν τις φάσεις και τα στάδια ανάπτυξης ενός ΠΣ. • Να υιοθετήσουν μια ευρύτερη οπτική για τα προβλήματα και τις λύσεις-προγράμματα χωρίς να εστιάζουν μόνο στο επίπεδο των τεχνικών χαρακτηριστικών, αλλά λαμβάνοντας υπόψη τον πραγματικό κόσμο (επιχειρήσεις, διοίκηση κλπ). 	<p>14. Πληροφοριακά Συστήματα [8 (7-9) + 0 Ω]</p> <p>14.1 Τύποι Πληροφοριακών Συστημάτων</p> <p>14.2 Μέρη – συνιστώσες ΠΣ (Υλικό - συστήματα, Λογισμικό, Δεδομένα, Διαδικασίες, Ανθρώπινο δυναμικό)</p> <p>14.3 Ο κύκλος ζωής ανάπτυξης ΠΣ</p> <p>14.4 Θέματα διαχείρισης ΠΣ</p> <ul style="list-style-type: none"> ▪ Γενικές αρχές στην ανάπτυξη ΠΣ ▪ Θέματα ασφάλειας πληροφοριών ▪ Ο ρόλος των βάσεων δεδομένων στα Πληροφοριακά Συστήματα Διοίκησης ▪ Ο ανθρώπινος παράγοντας στην αποδοχή ενός ΠΣ 	<ul style="list-style-type: none"> • Να αναζητήσουν εργαζόμενοι ομαδοσυνεργατικά Πληροφοριακά Συστήματα σε διάφορες περιοχές, όπως: <ul style="list-style-type: none"> ○ συστήματα υποστήριξης αποφάσεων ○ συστήματα διαχείρισης γνώσης • Να συζητηθούν τα πλεονεκτήματα χρήσης ΠΣ. • Να διακρίνουν τις βασικές οντότητες και τις σχέσεις μεταξύ τους σε ένα ΠΣ για μια θεματική περιοχή, όπου να αναδεικνύονται οι συνιστώσες του, όπως το Λογισμικό, το Υλικό, το ανθρώπινο δυναμικό και οι ρόλοι του. • Με παιχνίδι ρόλων να συνεργαστούν σε μια απλή μελέτη περίπτωσης για τη δημιουργία μίας μικρής μελέτης σκοπιμότητας για την ανάπτυξη ενός οικείου ΠΣ στους μαθητές.

Η διδακτική του μαθήματος γίνεται με βάση τον κοινωνικό εποικοδομισμό και τις σύγχρονες θεωρήσεις για την «επεξεργασία των πληροφοριών». Το μάθημα προτείνεται και πρέπει να γίνεται στο εργαστήριο πληροφορικής ακολουθώντας μεθόδους αναζήτησης και ανακάλυψης και οικοδόμησης της γνώσης. Ως κοινωνική οργάνωση της τάξης ακολουθείται και η ομαδοσυνεργατική προσέγγιση ιδιαίτερα σε θέματα αναζήτησης και ανάπτυξης εφαρμογών. Ο καθηγητής λειτουργεί ως διευκολυντής για την οικοδόμηση της γνώσης από τον μαθητή κινούμενος με τεχνικές του «πλαίσιου στήριξης» (scaffolding).

Οι γραπτές προαγωγικές εξετάσεις του μαθήματος γίνονται με τέσσερα (4) θέματα από την εξεταστέα ύλη, με τα οποία ελέγχεται η γνώση και η κατανόηση εννοιών και ορολογίας, η κριτική ικανότητα, η δυνατότητα αναπαραγωγής γνωστικών στοιχείων, η ικανότητα του μαθητή να αναλύει, να συνθέτει και να δημιουργεί, συνδυάζοντας γνώσεις και δεξιότητες που απέκτησε για την επίλυσή τους. Σημαντική θεωρείται και πρέπει να ακολουθείται η διαμορφωτική αξιολόγηση κατά τη διάρκεια της διδασκαλίας