

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΑΘΛΗΤΙΣΜΟΥ ΚΑΙ ΝΕΟΛΑΙΑΣ
ΔΙΕΥΘΥΝΣΗ ΜΕΣΗΣ ΤΕΧΝΙΚΗΣ ΚΑΙ ΕΠΑΓΓΕΛΜΑΤΙΚΗΣ ΕΚΠΑΙΔΕΥΣΗΣ
ΚΑΙ ΚΑΤΑΡΤΙΣΗΣ

ΕΝΙΑΙΕΣ ΓΡΑΠΤΕΣ ΤΕΛΙΚΕΣ ΠΡΟΑΓΩΓΙΚΕΣ ΚΑΙ
ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ

20 23 - 20 24

Γ΄ ΤΑΞΗΣ ΤΕΣΕΚ

ΣΕΙΡΑ Α'

ΘΕΩΡΗΤΙΚΗ ΚΑΤΕΥΘΥΝΣΗ

ΗΜΕΡΟΜΗΝΙΑ : Παρασκευή, 24 Μαΐου 2024

ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ: Εφαρμογές Προγραμματισμού III-TEM2

ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ : thips302

ΛΥΣΕΙΣ

ΜΕΡΟΣ Α΄: Αποτελείται από έξι (6) ερωτήσεις. Κάθε ορθή απάντηση βαθμολογείται με οκτώ (8) μονάδες.

1. Να χαρακτηρίσετε με σωστό (Σ) ή λάθος (Λ) καθεμιά από τις παρακάτω προτάσεις. Όλες οι προτάσεις αφορούν τη γλώσσα προγραμματισμού C++.

A/A	Πρόταση	Σωστό / Λάθος
α)	Ο μέγιστος αριθμός των συγκρίσεων στη σειριακή αναζήτηση (sequential search) είναι ίσος με το πλήθος των στοιχείων του πίνακα.	Σ
β)	Η δυαδική αναζήτηση (binary search) μπορεί να χρησιμοποιηθεί μόνο αν ο πίνακας είναι ήδη ταξινομημένος.	Σ
γ)	Ο αλγόριθμος της δυαδικής αναζήτησης (binary search) ξεκινά εξετάζοντας το «πρώτο» στοιχείο του πίνακα.	Λ
δ)	Ο αλγόριθμος της φουσαλίδας (bubble sort) ταξινομεί τα στοιχεία, χρησιμοποιώντας την αρχή της σύγκρισης και αντιμετάθεσης δύο γειτονικών στοιχείων.	Σ
ε)	Ο αλγόριθμος φουσαλίδας (bubble sort) είναι ένας από τους πιο αποδοτικούς αλγόριθμους ταξινόμησης.	Λ
στ)	Ο αλγόριθμος ταξινόμησης με εισαγωγή (insertion sort) είναι πιο αργός από τον αλγόριθμο ταξινόμησης φουσαλίδας (bubble sort).	Λ
ζ)	Οι αλγόριθμοι ταξινόμησης φουσαλίδας (bubble sort) και εισαγωγής (insertion sort) δεν μπορούν να ταξινομήσουν σε φθίνουσα σειρά αριθμούς.	Λ
η)	Ο αλγόριθμος σειριακής αναζήτησης (sequential search) μπορεί να χρησιμοποιηθεί μόνο για ακέραιους και πραγματικούς αριθμούς.	Λ

Κάθε σωστή απάντηση 1 μονάδα

2. Να χαρακτηρίσετε με σωστό (Σ) ή λάθος (Λ) καθεμιά από τις παρακάτω δηλώσεις μεταβλητών της γλώσσας προγραμματισμού C++.

A/A	Δήλωση μεταβλητών	Σωστό / Λάθος
(α)	<code>string names[3] = {"George", "Antonis"};</code>	Σ
(β)	<code>double points[3] = {20.5, 13.2, 45.3, 35.3};</code>	Λ
(γ)	<code>int points[3] [2]= {20, 13, 45};</code>	Σ
(δ)	<code>char characters[3] [2] = { };</code>	Σ

Κάθε σωστή απάντηση 2 μονάδες

3. Σας δίνεται το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++:

```
#include <iostream>
using namespace std;
int main(){
int i,j,pl=0,sum=0;


A


for (i=0;i<3;i++)
for (j=0;j<4;j++)
{
cout<<"Δώσε Αριθμό:";


B



Γ


}
for (i=0;i<3;i++)
for (j=0;j<4;j++)
{


Δ


pl++;
}
cout<<"Πλήθος θετικών τριψήφιων αριθμών:"<<pl<<endl;
cout<<"Άθροισμα όλων των στοιχείων του πίνακα:"<<sum<<endl;
return 0;
}
```

Να γράψετε:

(α) Την εντολή που πρέπει να τοποθετηθεί στη θέση A για να δηλωθεί ο δισδιάστατος πίνακας ακεραίων αριθμών, 3 γραμμών και 4 στηλών με το όνομα num.

int num[3][4];

(β) Την εντολή που πρέπει να τοποθετηθεί στη θέση B με την οποία να καταχωρίζει στον πίνακα num, τους δώδεκα (12) ακέραιους αριθμούς τους οποίους θα δίνει ο χρήστης.

cin >> num[i][j];

(γ) Την εντολή που πρέπει να τοποθετηθεί στη θέση Γ, έτσι ώστε να υπολογίζεται το άθροισμα όλων των στοιχείων του πίνακα num.

sum +=num[i][j];

(δ) Τη συνθήκη που πρέπει να τοποθετηθεί στη θέση Δ, έτσι ώστε να υπολογίζεται το πλήθος των θετικών τριψήφιων αριθμών που βρίσκονται στον πίνακα num.

if (num[i][j] >99 && num[i][j] <1000)

Κάθε σωστή απάντηση 2 μονάδες

4. Δίνεται ο πίνακας $\text{int arr}[3][3] = \{ \{0, 3, 6\}, \{5, 1\}, \{7, 2, -4\} \}$. Ποια θα είναι τα στοιχεία του πίνακα $\text{results}[3][3] = \{ 0 \}$ μετά την εκτέλεση των πιο κάτω εντολών της C++.

α) $\text{results}[2][0] = \text{arr}[2][2] + \text{arr}[1][2] * \text{arr}[2][2]$ (1,5 Mov.)

results[2][0] = -4 + 0 = -4

Σωστή απάντηση 1.5 Μονάδα

β) $\text{results}[2][1] = \text{arr}[0][2] \% \text{arr}[1][0] + \text{arr}[0][0]$ (1,5 Mov.)

results[2][1] = 1 + 0 = 1

Σωστή απάντηση 1.5 Μονάδα

γ) $\text{results}[1][1] = \text{arr}[2][0] + \text{arr}[0][1]$ (1,5 Mov.)

results[1][1] = 7 + 3 = 10

Σωστή απάντηση 1.5 Μονάδα

δ) $\text{results}[0][2] = \text{arr}[1][0] + \text{arr}[2][2] / \text{arr}[2][1]$ (1,5 Mov.)

results[0][2] = 5 + (-4) / (2) = 3

Σωστή απάντηση 1.5 Μονάδα

ε) Να συμπληρώσετε τον πιο κάτω πίνακα (results) με όλα τα στοιχεία του.

(2 Μον.)

	0	1	2
0	0	0	3
1	0	10	0
2	-4	1	0

Πίνακας results

2 Μονάδες η σωστή τοποθέτηση των μηδενικών και όλων των στοιχείων στον πίνακα - 0 μονάδες είτε η μη τοποθέτηση των μηδενικών είτε η λανθασμένη τοποθέτηση στοιχείων.

5. Να απαντήσετε τις πιο κάτω ερωτήσεις που αφορούν τον αντικειμενοστραφή προγραμματισμό.

(α) Τι ονομάζουμε κλάση (class);

(2 Μον.)

Κλάση ονομάζεται μια έννοια που μοντελοποιεί μια ιδέα, μια κατηγορία οντοτήτων, ένα πρότυπο ή ένα χαρακτηρισμό, και αποτελείται από δεδομένα (μεταβλητές) και συναρτήσεις (μεθόδους).

ή

Μια κλάση είναι ένας γενικός ορισμός αντικειμένου που καθορίζει χαρακτηριστικά και ενέργειες.

Σωστή απάντηση 2 Μονάδες

(β) Τι ονομάζουμε αντικείμενο (object);

(2 Μον.)

Αντικείμενο ονομάζεται ένα στιγμιότυπο μιας κλάσης και χαρακτηρίζεται από τις ιδιότητες του (μεταβλητές) και τις μεθόδους του (συναρτήσεις).

Σωστή απάντηση 2 Μονάδες

(γ) Τι είναι οι ιδιότητες (properties) και οι μέθοδοι (methods) στα αντικείμενα; (4 Μον.)

**Οι μέθοδοι είναι συναρτήσεις που ανήκουν σε ένα αντικείμενο στον αντικειμενοστραφή προγραμματισμό και εκτελούν μια συγκεκριμένη εργασία.
Οι ιδιότητες (μεταβλητές) δίνουν πληροφορίες για ένα αντικείμενο.**

1 Μονάδα για τον σωστό ορισμό των μεθόδων

1 Μονάδα για τον σωστό ορισμό των ιδιοτήτων

Σημείωση

Αποδεκτές όλες οι απαντήσεις που περιγράφουν τεκμηριωμένα τους ορισμούς.

6. Δίνεται η παρακάτω κλάση η οποία χρησιμοποιείται για τη διαχείριση των ATM διαφόρων τραπεζών:

```
class atm
{
private:
float balance;
bool extra;
public:
string trapeza;
    atm( )
    {
        balance = 3000;
    }
    void analipsi(float money)
    {
        if (money <= balance)
            balance = balance - money;
        else
            cout << " Το ποσό ανάληψης υπερβαίνει το υπόλοιπο του λογαριασμού!" << endl;
    }
    void katathesi(float money)
    {
        balance += money;
    }
    float vpoloipo()
    {
```

```
    return balance;
}
};
```

Να απαντήσετε στις πιο κάτω ερωτήσεις:

(α) Να αναφέρετε ένα ιδιωτικό μέλος της κλάσης atm: (1,5 Μον.)

balance, extra

Ένα από τα δυο

1,5 Μονάδα η σωστή απάντηση.

(β) Να γράψετε μια εντολή με την οποία δημιουργούνται δύο αντικείμενα της πιο πάνω κλάσης με ονόματα at1 και at2: (1,5 Μον.)

atm at1, at2;

1,5 Μονάδα η σωστή απάντηση.

(γ) Υποθέτουμε ότι υπάρχει ένα αντικείμενο της πιο πάνω κλάσης με όνομα at1. Να γράψετε μια εντολή με την οποία θα καταχωριστεί στη μεταβλητή-μέλος trapeza του αντικειμένου at1 το κείμενο «ΤΕΣΕΚbank»: (1,5 Μον.)

at1.trapeza="ΤΕΣΕΚbank" ;

1,5 Μονάδα η σωστή απάντηση.

(δ) Υποθέτουμε ότι υπάρχει ένα αντικείμενο της παραπάνω κλάσης με όνομα at1. Τι θα κάνει η εντολή at1.poso=1000; Αιτιολογήστε την απάντησή σας. (2 Μον.)

Θα υπάρξει λάθος κατά την εκτέλεση του προγράμματος διότι δεν υπάρχει μεταβλητή σαν δημόσιο μέλος της κλάσης με το όνομα poso.

1 Μονάδα η σωστή αναφορά του λάθους

1 Μονάδα η σωστή αιτιολόγηση

(ε) Υποθέτουμε ότι υπάρχει ένα αντικείμενο της παραπάνω κλάσης με όνομα at1. Να γράψετε μία εντολή με την οποία θα εμφανίζεται στην οθόνη το υπόλοιπο χρημάτων που υπάρχει στο at1. (1,5 Μον.)

cout << at1.υπολοιπο();

1,5 Μονάδα η σωστή απάντηση.

ΜΕΡΟΣ Β΄: Αποτελείται από τέσσερις (4) ερωτήσεις. Κάθε ορθή απάντηση βαθμολογείται με δέκα (10) μονάδες.

7. Δίδεται το πιο κάτω πρόγραμμα, στη γλώσσα προγραμματισμού C++. Το πρόγραμμα καταχωρίζει τα ονόματα δεκαπέντε (15) επιλεγόμενων μαθημάτων σε ένα μονοδιάστατο πίνακα με το όνομα `epilegomena` και το πλήθος των μαθητών που επέλεξαν το κάθε ένα από αυτά σε ένα άλλο παράλληλο μονοδιάστατο πίνακα με το όνομα `plith`. Ακολουθώς ταξινομεί τους δύο πίνακες σε αύξουσα σειρά με βάση το πλήθος, χρησιμοποιώντας τον αλγόριθμο εισαγωγής (`insertion sort`).

```
#include <iostream>
#define N 15
using namespace std;
int main ( ) {
```

A

```
for (i=0; i<N; i++)
    cin>>epilegomena[i]>>plith[i];
```

```
for (i=1;i<N;i++) {
    temp=epilegomena[i];
    tempa=plith[i];
    k=i-1;
```

B

```
{
    plith[k+1]=plith[k];
    epilegomena[k+1]=epilegomena[k];
    k--;
}
```

Γ

```
}
for (i=0;i<N;i++)
    cout<<epilegomena[i]<<" "<<plith[i]<<endl;
return 0; }
```

Να γράψετε:

- (α) Την/τις εντολή/ές που πρέπει να τοποθετηθεί/ούν στη θέση A έτσι ώστε να δηλώνονται οι μεταβλητές του προγράμματος. (3 Μον.)

string epilegomena[N],temp; 0,5 Μονάδα για κάθε σωστή μεταβλητή
int plith[N], k, tempa, i; 0, 5 Μονάδα για κάθε σωστή μεταβλητή

(3 Μονάδες συνολικά)

(β) Την/τις εντολή/ές που πρέπει να τοποθετηθεί/ούν στη θέση Β έτσι ώστε οι παράλληλοι πίνακες *epilegomena* και *plith* να ταξινομούνται σε αύξουσα σειρά (με βάση το πλήθος των μαθητών). (3 Μον.)

while(k>=0 && plith[k]>tempa) 3 Μονάδες για την σωστή συνθήκη

(γ) Την/τις εντολή/ές που πρέπει να τοποθετηθεί/ούν στη θέση Γ ώστε να υλοποιηθεί η ταξινόμηση των δύο παράλληλων πινάκων *epiligomena* και *plith*. (4 Μον.)

plith[k+1]=tempa; 2 Μονάδες
epilegomena[k+1]=temp; 2 Μονάδες

(4 Μονάδες Συνολικά)

8. Να γράψετε συνάρτηση στην γλώσσα προγραμματισμού C++ που να επιστρέφει πόσες φορές εμφανίζεται ένας χαρακτήρας σε ένα πίνακα χαρακτήρων με την μέθοδο της σειριακής αναζήτησης (*sequential search*). Ο πίνακας, το μέγεθος του πίνακα και ο χαρακτήρας να δίνονται ως παράμετροι στην συνάρτηση.

1 Μονάδα ο σωστός ορισμός της συνάρτησης,

1,5 Μονάδα ο σωστός ορισμός των παραμέτρων της συνάρτησης,

int sequential_search(char arr [], int N, char target)

{	0,25 Μονάδα
int count=0;	1 Μονάδα
for (int i=0; i<N; i++)	1 Μονάδα
if (target == arr[i])	2 Μονάδες
count +=1;	2 Μονάδες
return count;	1 Μονάδα
}	0,25 Μονάδα

Σημείωση: Η λύση είναι ενδεικτική. Οποιαδήποτε άλλη επιστημονικά τεκμηριωμένη λύση είναι αποδεκτή.

9. Πιο κάτω παρουσιάζονται δύο συναρτήσεις στη γλώσσα προγραμματισμού C++. Η πρώτη συνάρτηση εκτελεί ταξινόμηση μονοδιάστατου πίνακα πραγματικών αριθμών κατά αύξουσα σειρά χρησιμοποιώντας τον αλγόριθμο της φυσαλίδας (*Bubble Sort*) και η δεύτερη εκτελεί αναζήτηση ενός πραγματικού αριθμού και επιστρέφει την θέση του στον ταξινομημένο πίνακα χρησιμοποιώντας τον αλγόριθμο δυαδικής αναζήτησης (*Binary Search*). Αν δεν εντοπιστεί ο αριθμός επιστρέφει την τιμή -1. Στον πίνακα δεν υπάρχει αριθμός που να επαναλαμβάνεται. Στις συναρτήσεις υπάρχουν τέσσερα (4)

λογικά λάθη. Να τα εντοπίσετε και να τα διορθώσετε συμπληρώνοντας στον πιο κάτω πίνακα τον αριθμό της γραμμής και την διορθωμένη εντολή. Στις συναρτήσεις να μη γίνει καμία προσθήκη ή αφαίρεση εντολής.

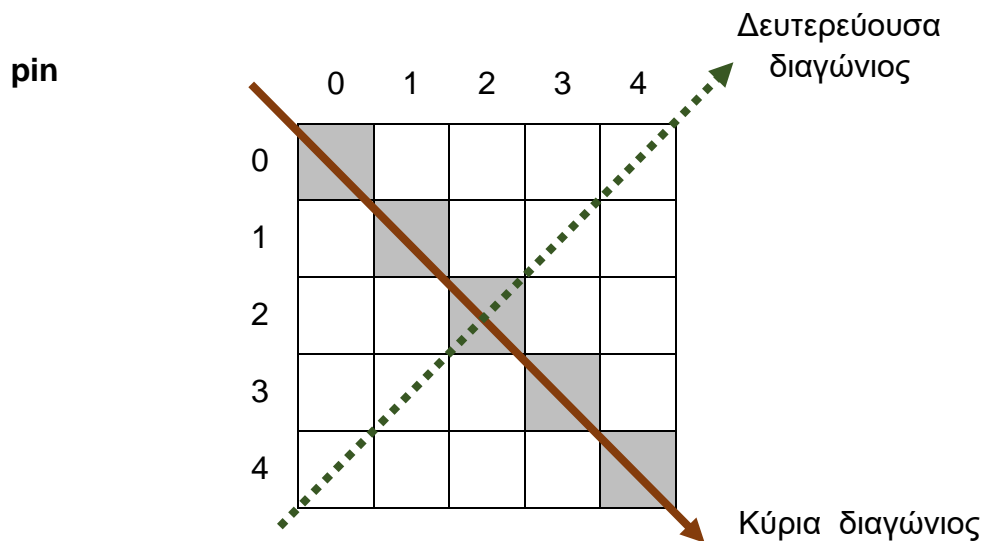
```
1 void bubbleSort(double arr[ ], int N) {
2     double temp;
3     for (int i=0; i<N-1; i++) {
4         for (int j=i+1; j<N; j++) {
5             if (arr[ i ]<arr[ j ]) {
6                 arr[ i ] = temp;
7                 arr[ i ] = arr[ j ];
8                 arr[ j ] = temp;
9             }
10        }
11    }
12 }
13 int binarySearch(double arr[ ], int left, int right, double key) {
14     while (left < right) {
15         int mid = right + left ;
16         if (arr[mid] == key)
17             return mid;
18         else if (key < arr[mid])
19             right = mid - 1;
20         else
21             left = mid + 1;
22     }
23     return -1;
24 }
```

A/A	Αριθμός Γραμμής	Διορθωμένη εντολή
1	5	if (arr [i] > arr [j]) {
2	6	temp = arr [i] ;
3	14	while (left <= right) {
4	15	int mid = (right + left) / 2;

1,25 ο σωστός εντοπισμός του κάθε λάθους, 1,25 η σωστή διόρθωση του κάθε λάθους

10. Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού C++ το οποίο:

- (α) Να ζητά **25 ακέραιους** αριθμούς και να τους αποθηκεύει σε ένα τετραγωνικό πίνακα **5X5** με το όνομα **pin**. Οι αριθμοί να αποθηκεύονται κατά στήλη, δηλαδή να γεμίζει πρώτα η 1^η στήλη του πίνακα, μετά η 2^η στήλη κ.τ.λ.
- (β) Να υπολογίζει και να επιστρέφει τον μεγαλύτερο αριθμό της κύριας διαγώνιου με τη βοήθεια της συνάρτησης **smax**. Η συνάρτηση **smax** θα δέχεται ως παράμετρο τον πίνακα **pin** από την κύρια συνάρτηση (κυρίως πρόγραμμα) και θα υπολογίζει και θα επιστρέφει στη κύρια συνάρτηση τον μεγαλύτερο αριθμό της κύριας διαγώνιού. Ο αριθμός αυτός θα τυπώνεται από τη κύρια συνάρτηση (κυρίως πρόγραμμα).



Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εξαγωγή των αποτελεσμάτων σύμφωνα με το πιο κάτω παράδειγμα:

Παράδειγμα Εξόδου

(οθόνη)

Δώσε 25 ακέραιους αριθμούς:

Μέγιστος αριθμός κύριας διαγώνιού:

```

#include <iostream>
using namespace std;
int smax(int array[5][5])  ή  int smax(int array[ ][5])
{
    int maxNumber=array[0][0];
    for (int i=1;i<5;i++)
    {
        if (array[i][i] >= maxNumber)  ή  ( if (array[i][i] > maxNumber) )
            maxNumber=array[i][i];
    }
    return maxNumber;
}
int main()
{
    int pin[5][5];
    cout << "Δώσε 25 ακέραιους αριθμούς: " << endl;
    for (int stili=0;stili<5;stili++)
    {
        for (int grammi=0;grammi<5;grammi++)
        {
            cin >> pin[grammi][stili];
        }
    }
    cout << "Μέγιστος αριθμός κύριας διαγώνιου: " << smax(pin);
}

```

4 Μονάδες

5 Μονάδες

1 Μονάδα

Αποδεκτή λύση για την συνάρτηση είναι και η εξής:

```
int smax(int array[5][5])  ή  int smax(int array[ ][5])
{
int maxNumber=array[0][0];
for (int i=1; i<5; i++)
    for(int j=1; j<5; j++) {
        if ( i == j && array[i][j] > maxNumber )
            maxNumber=array[i][j];}
return maxNumber;
}
```

Αφαιρετικά (γενικά)

- 0,25 αν λείπει το #include <iostream>
- 0,25 αν λείπει το using namespace std;
- 0,25 αν λείπει το int main ()
- 0,25 αν λείπει { ή και }
- 0,25 αν λείπουν δύο η περισσότερα ; (μια φορά)

Σημείωση: Η λύση είναι ενδεικτική. Οποιαδήποτε άλλη επιστημονικά τεκμηριωμένη λύση είναι αποδεκτή.

ΜΕΡΟΣ Γ΄: Αποτελείται από μία (1) ερώτηση. Η ορθή απάντηση βαθμολογείται με δώδεκα (12) μονάδες.

11. Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού C++ το οποίο :

(α) Να δημιουργεί μια κατηγορία (κλάση), με όνομα Student για μαθητές. (2 Μον.)

(β) Στα ιδιωτικά πεδία της κλάσης Student, να περιλαμβάνονται τα ακόλουθα: (2 Μον.)

- i. Ονοματεπώνυμο μαθητή
- ii. Τμήμα μαθητή
- iii. Όνομα Μαθήματος
- iv. Ένας μονοδιάστατος πίνακας τριών (3) ακεραίων τιμών όπου θα καταχωρούνται οι βαθμοί που αφορούν τις βαθμολογίες του μαθητή στο πρώτο τετράμηνο, στο δεύτερο τετράμηνο και στην τελική εξέταση.

(γ) Στα δημόσια πεδία της κλάσης Student, να περιλαμβάνονται τα ακόλουθα: (6 Μον.)

- i. Ένας κατασκευαστής (constructor), ο οποίος θα ενημερώνει τα πεδία της κλάσης με τιμές που θα του παρέχονται ως παράμετροι κατά τη δημιουργία του.
- ii. Μια μέθοδος με όνομα mesos() που θα υπολογίζει τον μέσο όρο των βαθμών του μαθητή και θα τον επιστρέφει στο κυρίως πρόγραμμα (συνάρτηση main).
- iii. Μια μέθοδος με όνομα details() που θα εμφανίζει όλα τα στοιχεία του μαθητή.

(δ) Στη κύρια συνάρτηση να υλοποιείται ένα αντικείμενο της κλάσης Student με τα ακόλουθα χαρακτηριστικά: (1 Μον.)

Όνοματεπώνυμο: Γιώργος Γεωργίου

Τμήμα: ΘΗΨ3

Όνομα Μαθήματος: Εφαρμογές Προγραμματισμού III

Βαθμοί: 15, 10, 20

(ε) Η κύρια συνάρτηση με τη βοήθεια των μεθόδων της κλάσης Student να εμφανίζει όλα τα στοιχεία του μαθητή και τον μέσο όρο της βαθμολογίας του μαθητή με δύο δεκαδικά ψηφία. (1 Μον.)

Παράδειγμα εξόδου

Όνοματεπώνυμο: Γιώργος Γεωργίου

Τμήμα: ΘΗΨ3

Όνομα Μαθήματος: Εφαρμογές Προγραμματισμού III

Μέσος όρος Βαθμολογίας: 15.00

```
#include <iostream>
#include <iomanip>
using namespace std;
```

```
class Student {
    private:
        string name;
        string department;
        string courseName;
        int grades[3]; // Βαθμοί για τα 2 τετράμηνα και τελική εξέταση
    public:
        // Κατασκευαστής
        Student(string n, string dep, string course, int grade1,
int grade2, int grade3) {
            name = n;
            department = dep;
            courseName = course;
            grades[0] = grade1;
            grades[1] = grade2;
            grades[2] = grade3;
        }
        // Μέθοδος που επιστρέφει τα στοιχεία του μαθητή
        void details() {
            cout << "Όνοματεπώνυμο: " << name << endl;
            cout << "Τμήμα: " << department << endl;
            cout << "Όνομα Μαθήματος: " << courseName << endl;
        }
        // Μέθοδος που επιστρέφει τον μέσο όρο της βαθμολογίας
        double mesos() {
            int sum = 0;
            for (int i = 0; i < 3; ++i) {
                sum += grades[i];
            }
            return float (sum) / 3; }
}
```

1 Μονάδες

2 Μονάδες

2 Μονάδες

2 Μονάδες

2 Μονάδες

};

1 Μονάδα

```
int main ( ) {
```

```
    // Δημιουργία αντικειμένου Student
```

```
    Student student1("Γιώργος Γεωργίου", "ΘΗΨ3",
```

1 Μονάδα

```
"Εφαρμογές Προγραμματισμού ΙΙΙ", 15, 10, 20);
```

```
    // Εμφάνιση στοιχείων μαθητή και μέσου όρου
```

```
    student1.details();
```

0,5 Μονάδα

```
    cout << "Μέσος Όρος Βαθμολογίας: " <<
```

```
fixed<<setprecision(2)<<student1.mesos() << endl;
```

0,5 Μονάδα

```
    return 0;
```

```
}
```

Αφαιρετικά (γενικά)

-0,25 αν λείπει το #include <iostream>

-0,25 αν λείπει το #include <iomanip>

-0,25 αν λείπει το using namespace std;

-0,25 αν λείπει το int main ()

-0,25 αν λείπει { ή και }

-0,25 αν λείπουν δύο ή περισσότερα ; (μια φορά)

Σημείωση: Η λύση είναι ενδεικτική. Οποιαδήποτε άλλη επιστημονικά τεκμηριωμένη λύση είναι αποδεκτή.