

1)Πόσο θα είναι το i,j μετά από την εκτέλεση του παρακάτω κώδικα;

```
int i=5;
```

```
int j=i++;
```

a)5,5

b)5,6

**c)6,5 Σωστό**

d)6,6

2)Έστω i=1, j=2, k=3 και m=2. Τι θα τυπώσουν οι παρακάτω δηλώσεις;

```
cout << (j ==3) <<endl;
```

```
cout << (m<=99 && k < m) <<endl;
```

```
cout <<(!m) <<endl;
```

```
cout <<(!(k > m)) << endl;
```

a>true, true, true, false

**b)0,0,0,0 Σωστό**

c)0,0,1,0

d>true, false, 1, 0

3)Τι τυπώνει ο παρακάτω κώδικας;

```
#include <iostream>
```

```
#include <memory>
```

```
class Resource
```

```
{
```

```
public:
```

```
Resource() { std::cout << "Constructor\n"; }
```

```
~Resource() { std::cout << "Destructor\n"; }
```

```
};
```

```
int main()
```

```
{
```

```
std::unique_ptr<Resource> res ( new Resource() );
```

```
return 0;
```

```
}
```

**a) Constructor**

**Destructor Σωστό**

b) Constructor

c) Destructor

d)Τίποτα από τα παραπάνω

4) Έστω i=1, j=2, k=3 και m=2. Τι θα τυπώσουν οι παρακάτω δηλώσεις;

```
cout << (i==1) << endl;
```

```
cout << (i >=1 && j < 4) << endl;
```

```
cout << (k+m < j || 3-j >=k) << endl;
```

```
cout << (!(j-m)) << endl;
```

a) 1, 1, true, false

b) true, true, false, false

c) 1, 0, 0, 1

**d) 1, 1, 0, 1 Σωστό**

5) Ποια είναι η σωστή δήλωση του prefix increment operator για ένα αντικείμενο της κλάσης Date;

**a) Date &operator++(); Σωστό**

b) Date operator++( int );

c) Date &operator+=( unsigned int );

d) Date &operator++(int);

6) Τι τυπώνει ο παρακάτω κώδικας;

```
#include <iostream>
```

```
using namespace std;
```

```
class Shape {
```

```
public:
```

```
int width =1;
```

```
int height = 2;
```

```
};
```

```
class Rectangle: public Shape{
```

```
public:
```

```
int side =3;
```

```
};
```

```
int main() {
```

```
Rectangle *r = new Rectangle;
```

```
Shape *s = r;
```

```
s->width = 4;
```

```
s = r;
```

```
cout << r->width << endl;
```

```
cout << s->width << endl;
```

```
return 0;
```

```
}
```

a) 1 1

b) 2 2

c) 3 3

**d) 4 4 Σωστό**

7) Ποια είναι η σωστή δήλωση του postfix increment operator για ένα αντικείμενο της κλάσης Date;

a) Date &operator++();

**b) Date operator++( int ); Σωστό**

c) Date &operator+=( unsigned int );

d) Date &operator++(int);

8) Τι τυπώνει ο παρακάτω κώδικας;

```
#include <iostream>
```

```
#include <memory>
```

```
class Resource
```

```
{
```

```
public:
```

```
Resource() { std::cout << "Constructor\n"; }
```

```
~Resource() { std::cout << "Destructor\n"; }
```

```
};
```

```
int main()
```

```
{
```

```
Resource* res = new Resource();
```

```
return 0;
```

```
}
```

a) Constructor

Destructor

**b) Constructor Σωστό**

c) Destructor

d) Τίποτα από τα παραπάνω

9) Τι τυπώνει ο παρακάτω κώδικας;

```
#include <iostream>

using namespace std;

int main()

{

    unsigned int y = 0;

    unsigned int x = 1;

    unsigned int total = 0;

    while ( x <= 10 )

    {

        y=x*x;

        cout <<y << endl;

        total += y;

        ++x;

    }

    cout << "Total is " << total << endl;

}
```

a) Τα τετράγωνα των αριθμών από 0 έως και 10 και το άθροισμα των τετραγώνων.

**b) Τα τετράγωνα των αριθμών από 1 έως και 10 και το άθροισμα των τετραγώνων. Σωστό**

c) Το άθροισμα των τετραγώνων των αριθμών από 1 έως και 10.

d) Τα τετράγωνα των αριθμών από 1 έως και 10.

10) Τι θα τυπώσει ο παρακάτω κώδικας;

```
#include<iostream>

using namespace std;

class Base {

public:

    virtual void show() { cout<<" In Base \n";};

};

class Derived: public Base {

public:

    void show() { cout<<"In Derived \n"; };

}
```

};

```
int main(void) {  
    Base *bp = new Derived;  
    bp->show();  
    return 0;  
}
```

a) In Base

**b) In Derived Σωστό**

c) In Derived

In Base

d) In Base

In Derived

11)Object slicing προκαλεί:

a)Εκχώρηση αντικειμένου τύπου κλάσης βάσης σε αντικείμενο της παραγόμενης κλάσης(με τιμή).

**b)Εκχώρηση αντικειμένου παραγόμενης κλάσης σε αντικείμενο τύπου κλάσης βάσης (με τιμή). Σωστό**

c) Εκχώρηση αντικειμένου παραγόμενης κλάσης σε αντικείμενο τύπου κλάσης βάσης (με αναφορά).

d) Εκχώρηση αντικειμένου τύπου κλάσης βάσης σε αντικείμενο της παραγόμενης κλάσης(με αναφορά).

12) Ποιο σύμβολο χρησιμοποιείται για τον ορισμό κλάσης εκτός εμβέλειάς της;

a) :

**b) :: Σωστό**

c) #

d) ~

13)Όταν δημιουργείται ένα αντικείμενο με new:

a) Δεσμεύεται χώρος στη στοίβα(stack)

b) Δεσμεύεται χώρος στο σωρό(heap) και αποδεσμεύεται αυτόματα όταν βγει εκτός εμβέλειας

c) Δεσμεύεται χώρος στη στοίβα(stack) και αποδεσμεύεται αυτόματα όταν βγει εκτός εμβέλειας

**d)Δεσμεύεται χώρος στο σωρό(heap) Σωστό**

14)Τι είναι προτιμότερο όταν κάνουμε catch μια εξαίρεση e που δεν γνωρίζουμε τον ακριβή τύπο της;

a) catch (Exception e),γιατί η what() είναι πολυμορφική

**b)catch (Exception &e),γιατί η what() είναι πολυμορφική Σωστό**

c) catch (Exception e),γιατί η what() δεν είναι πολυμορφική

d) catch (Exception &e),γιατί η what() δεν είναι πολυμορφική

15) Τι τυπώνει ο παρακάτω κώδικας;

```
#include <iostream>

using namespace std;

class Shape {

public:

int width = 1;

int height = 2;

virtual void getType(){

cout << "Shape"<<endl;}

};

class Rectangle: public Shape {

public:

int side = 3;

void getType () override {

cout << "Rectangle"<<endl;

}

};

int main() {

Shape* s ;

Rectangle r ;

s = &r;

s->getType();

r.getType();

return 0;

}
```

a) Shape

Shape

b) Rectangle

Rectangle Σωστό

c)Rectangle

Shape

d) Shape

Rectangle

16) Τι τυπώνει ο παρακάτω κώδικας;

```
#include <iostream>

using namespace std;
```

```

class Shape {
public:
int width    = 1;
int height   = 2;
};

class Rectangle: public Shape {
public:
int side =3;
};

int main() {
Rectangle r;
Shape s;
s.width = 4;
s = r;
cout << r.width << endl;
cout << s.width << endl;
return 0;
}

```

a) 1

4

b) 1

3

c) 1

**1 Σωστό**

d) 1

2

17) Ποιο είναι σωστό από τα παρακάτω;

a) Κάθε κλάση έχει vTable

**b) vptr είναι ένα δείκτης για το vTable Σωστό**

c) Κάθε αντικείμενο έχει vptr

d) Όλα τα παραπάνω

18) Μια friend συνάρτηση:

a) μπορεί να έχει πρόσβαση μόνο σε public members μιας κλάσης

**b) μπορεί να έχει πρόσβαση μόνο σε private members μιας κλάσης Σωστό**

c) μπορεί να έχει πρόσβαση μόνο σε protected members μιας κλάσης

d) μπορεί να έχει πρόσβαση μόνο σε όλα members μιας κλάσης

19)Τι από τα παρακάτω ισχύει για ένα αντικείμενο που δημιουργείται δυναμικά;

**a)Ο τελεστής delete καλεί τον destructor και απελευθερώνει τη μνήμη. Σωστό**

b)Ο destructor καλεί τον τελεστή delete και απελευθερώνει τη μνήμη.

c)Ο destructor απελευθερώνει τη μνήμη.

d) Ο destructor καλεί τον τελεστή delete και η μνήμη παραμένει δεσμευμένη.

20)Στην ιεραρχία κλάσεων Point3/Circle4/Cylinder

a) Ο Point3 constructor,καλείται πρώτος και ολοκληρώνει την εκτέλεσή του τελευταίος.

b) Ο Cylinder constructor,καλείται πρώτος και ολοκληρώνει την εκτέλεσή του τελευταίος.

**c)Ο Point3 constructor,καλείται τελευταίος και ολοκληρώνει την εκτέλεσή του πρώτος. Σωστό**

d) Ο Cylinder constructor,καλείται τελευταίος και ολοκληρώνει την εκτέλεσή του πρώτος.

21)Ποιο από τα παρακάτω είναι η σωστή σύνταξη για μια πρότυπη συνάρτηση;

a)template<class T>return\_type Function\_name(parameters)

b) template<typename T>return\_type Function\_name(parameters)

**c)Και τα 2 Σωστό**

d)Κανένα από τα παραπάνω

22)Abstract κλάση ονομάζεται αυτή που έχει

a)Μια virtual συνάρτηση

b)Όλες τις συναρτήσεις της virtual

**c)Μια pure virtual συνάρτηση Σωστό**

d)Όλες τις συναρτήσεις της pure virtual

23)Έστω κλάση time με πεδία int hour, minute, second.Ποιος είναι ένα κατάλληλος constructor για την αρχικοποίηση των πεδίων σε 0;

a) Time::Time() : hour(0),minute(0),second(0){};

b) Time::Time() : hour(0),minute(0){second = 0};

**c)Time::Time(int a=0,int b=0,int c=0){}; Σωστό**

d)Ο 1<sup>ος</sup> και ο 2<sup>ος</sup>

24)Ο constructor της παραγόμενης κλάσης καλεί τον constructor της κλάσης βάσης:

a)Άμεσα, μέσα στο σώμα του constructor

b)Άμεσα, με super

**c)Άμεσα,στην member initialization list Σωστό**

d)Μπορεί να καλέσει μόνο έμμεσα, τον default



25)\_\_\_\_\_ χρησιμοποιείται για προγραμματισμό με γενικεύσεις

**a) Η κληρονομικότητα Σωστό**

b)Οι εικονικές συναρτήσεις

c)Τα πρότυπα

d)Κανένα από τα παραπάνω

26)Μπορούμε να έχουμε πολυμορφισμό όταν χρησιμοποιούμε:

a)Δείκτη κλάσης βάσης σε αντικείμενο της κλάσης βάσης

b)Δείκτη παραγόμενης κλάσης σε αντικείμενο της παραγόμενης κλάσης

**c)Δείκτη κλάσης βάσης σε αντικείμενο της παραγόμενης κλάσης Σωστό**

d)Δείκτη παραγόμενης κλάσης σε αντικείμενο της κλάσης βάσης

27)Ποια από τα παρακάτω είναι σωστή δήλωση ενός array;

**a)int array[10]; Σωστό**

b)int array;

c)array{10};

d)array array[10];

28)Η σύνδεση του κώδικα με τυχόν βιβλιοθήκες που χρησιμοποιεί γίνεται από:

a)Τον compiler

b)Τον preprocessor

**c)Τον linker Σωστό**

d)Τη CPU

29) Ποιος είναι ο αριθμός ευρετηρίου του τελευταίου στοιχείου ενός πίνακα με 9 στοιχεία;

a)9

**b)8 Σωστό**

c)0

d) Καθορίζεται από τον προγραμματιστή

30) Ποιο από τα παρακάτω είναι σωστή δήλωση κλάσης;

**a)class A { int x; }; Σωστό**

b)class B { }

c)public class A { }

d)object A { int x; };

31) Ποια είναι η σωστή δήλωση για μια συνάρτηση που δέχεται σαν όρισμα ένα δείκτη float, ένα δείκτη σε δείκτη char και επιστρέφει ένα δείκτη σε δείκτη int;

a) `int ** fun (float **, char **)`

b) `int * fun (float *, char *)`

c) `int ** fun (float *, char **)` Σωστό

d) `int *** fun (* float, ** char)`

?Unknown? 32) Η εντολή `cout << names.at (3)` προϋποθέτει:

a) `vector names = {"A", "B", "C", "D"};`

b) `vector <string> names = {"A", "B", "C", "D"};`

c) `string names [] = {"A", "B", "C", "D"};`

d) Το 2<sup>ο</sup> ή το 3<sup>ο</sup>

?Unknown? 33) Έστω ότι θέλουμε να κάνουμε overload τη συνάρτηση `cube`. Υπάρχει κάποιο πρόβλημα στον κώδικα;

`double cude(int);`

`int cube(int);`

a) Ναι, μόνο η αλλαγή επιστρεφόμενου τύπου δεν συνιστά overload

b) Όχι, γιατί αρκεί να είναι ίδιος ο τύπος της παραμέτρου

c) Ναι, γιατί ο τύπος της παραμέτρου είναι ο ίδιος

d) Όχι, γιατί επιστρέφουν διαφορετικό τύπο

?Unknown? 34) Όταν υπερφορτώνουμε έναν δυαδικό τελεστή ως μέλος κλάσης:

a) Το αριστερό μέλος του τελεστή είναι το αντικείμενο για το οποίο καλείται

b) Το δεξιό μέλος του τελεστή είναι το αντικείμενο για το οποίο καλείται

c) Το αριστερό μέλος του τελεστή είναι το όρισμα της συνάρτησης

d) Θα πρέπει να δηλωθεί ως friend

?Unknown? 35) Έστω η παρακάτω πρότυπη κλάση. Ποιο από τα παρακάτω είναι έγκυρη δημιουργία στιγμιότυπου της κλάσης;

Εδώ είχε μια και εικόνα με την πρότυπη κλάση.
---

a) `mypair<int, int> myobject (1, 2);`

b) `mypair<> myobject (1, 2);`

c) `mypair<int> myobject (1, 2);`

d) `mypair<int, int> myobject (1);`



