

Εισαγωγή στο Λογισμικό II (Γλώσσα Java)

Ιούνιος 2000

- 1) Δώστε ένα παράδειγμα χρήσης (1-2 γραμμές) των παρακάτω τελεστών και εξηγήστε (1-2 γραμμές) τον τρόπο χρήσης τους: `&&`, `||`, `&`, `|`, `<<`, `>>` και `>>>`. Αν κάποιος τελεστής έχει παραπάνω από μία χρήση περιγράψτε όλες τις χρήσεις του.

- 2) Εξετάστε το παρακάτω κομμάτι προγράμματος:

```
for (a = 0 ; a < 3 ; a++) {  
    if (a == 1) continue;  
    for (b = 0 ; b < 3 ; b++) {  
        if (b == 1) break;  
        System.out.println(a + ", " + b);  
    }  
}
```

Ποιές από τις παρακάτω γραμμές θα εμφανιστούν στην έξοδο; Επιλέξτε όλες όσες θα εμφανιστούν

- a) 0, 0
- b) 0, 1
- c) 0, 2
- d) 1, 0
- e) 1, 1
- f) 1, 2
- g) 2, 0
- h) 2, 1
- i) 2, 2

- 3) Εξετάστε τον παρακάτω κώδικα:

```
public class Quiz {  
    public static void main(String[] args) {  
        String a = "Hello ";  
        String b = "World!";  
        f(a,b);  
        System.out.println(a);  
    }  
    public static void f(String x, String y) {  
        x += y;  
    }  
}
```

Ποια από τις παρακάτω προτάσεις περιγράφει την συμπεριφορά του παραπάνω προγράμματος:

- a) Η έξοδος είναι: Hello
 - b) Η έξοδος είναι: World!
 - c) Η έξοδος είναι: Hello World!
 - d) Ο compiler απορρίπτει την πρόταση `x += y` επειδή δεν μπορούμε να αυξήσουμε την τιμή ενός String αντικειμένου.
- 4) Εξηγήστε την χρήση των λέξεων **abstract**, **final**, **native**, **static**, **synchronized** όταν χρησιμοποιούνται στον ορισμό μίας μεθόδου τάξης (class).

- 5) Περιγράψτε με χρήση ενός πίνακα τις διαφορετικές δυνατότητες προσπέλασης στα μέλη μίας τάξης. Να συμπληρώσετε και να επεκτείνετε τον παρακάτω πίνακα:

Κατάσταση	Public	Protected	Default (or package)	Private
Ίδιο πακέτο, όχι sub-class	Ναι			

- 6) Περιγράψτε με ένα παράδειγμα το φαινόμενο της επικάλυψης (overriding) και της υπερφόρτωσης (overloading) μεθόδων τάξης.
- 7) Στον παρακάτω κώδικα γράψτε ποια θα είναι η έξοδος του και δικαιολογήστε την απάντησή σας.
package exercise.inheritance

```
public class Test {
    public static void main(String[] args) {
        SuperClass y = new SuperClass();
        SubClass x = new SubClass();
        x.g();
    }
}

class SuperClass {
    public SuperClass() {
        f();
    }
    public void f() {
        System.out.println("SuperClass.f() called");
    }
    public void g() {
        f();
    }
}

class SubClass extends SuperClass {
    public void f() {
        System.out.println("SubClass.f() called");
    }
}
```

- 8) Περιγράψτε με ένα παράδειγμα τον τρόπο χρήσης του όρου **try-catch-finally** στα προγράμματα.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ ΚΑΙ ΚΑΛΟ ΚΑΛΟΚΑΙΡΙ