ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
Κ10: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ
Εξετάσεις 31 Ιανουαρίου 2017

1. (α′) Δώστε έναν (τουλάχιστον) λόγο για τον οποίο η γλώσσα Java μπορεί να θεωρηθεί περισσότερο "αντικειμενοστραφής " από την γλώσσα C++.

   (β′) Δίδεται το παρακάτω πρόγραμμα Java:

```
public class MainClass {
    public static void main(String[] args) {
    A a1 = new A(); A a2 = new A(10);
    B b1 = new B(); B b2 = new B(100);
    System.out.println("////////////////////////////////////");
    a1.print(); a2.print(); b1.print(); b2.print();
    System.out.println("////////////////////////////////////");
    a1 = b1; a1.print(); b1 = b2; a1.print(); } }

class A{
    private int data;
        { System.out.println("A new A was just created"); }
    public A() {
            System.out.println("The A data is " + data); }
    public A(int d) { data = d;
            System.out.println("The A data is " + data); }
    public void print() { System.out.println("The A data is " + data); } }

class B extends A{
    private int data;
        { System.out.println("A new B was just created"); }
    public B() { System.out.println("The B data is " + data); }
    public B(int d) { data = d;
                System.out.println("The B data is " + data); }
    public void print() { super.print();
          System.out.println("The B data is " + data); } }
```

   Επίσης, δίδεται και το παρακάτω πρόγραμμα C++:

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

class A{ int data;
    public:
       A(int d = 0) {
         cout << "A new A was just created" << endl;
             data = d;
         cout << "The A data is " << data << endl ;}
    void print() { cout << "The A data is " << data << endl ; } };

class B : public A{ int data;
    public:
      B(int d = 0) {
        cout << "A new B was just created" << endl;
        data = d;
```

```
            cout << "The B data is " << data << endl ; }
        void print() {
              A::print();
              cout << "The B data is " << data << endl; } };

int main() {
    A a1; A a2(10); B b1; B b2(100);
    cout << "////////////////////////////////////" << endl;
    a1.print(); a2.print(); b1.print(); b2.print();
    cout << "////////////////////////////////////" << endl;
    a1 = b1; a1.print(); b1 = b2; a1.print(); }
```

Τα προγράμματα δείχνουν παρόμοια, όμως, δεν έχουν τα ίδια αποτελέσματα. Αιτιολογώντας την απάντησή σας, δώστε τα αποτελέσματα του καθενός. Επίσης, εξηγήσετε συνοπτικά πού και γιατί τα αποτελέσματα διαφέρουν.

2. Αιτιολογώντας την απάντησή σας, δώστε το αποτέλεσμα της εκτέλεσης του παρακάτω προγράμματος C++:

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

class A{ int data;
    public:
      A(int i=10):data(i)
        { cout << "An A was just constructed with " << i << endl; }
      ~A(){ cout << "An A will be destroyed with " << data << endl; }
      void inc() { data++; }
      int get() { return data; }
      void print() { cout << "The A data is: " << data << endl; } };

class B1 { A data;
    public:
      B1(const A& a): data(a)
         { cout << "A B1 was just constructed " << endl; }
      ~B1() { cout << "A B1 will be destroyed with " << data.get() << endl; }
      B1 process() { data.inc(); return *this; }
      void print() { cout << "The B1 data is: " << data.get() << endl ; } };

class B2 { A& data;
    public:
      B2(A& a): data(a)
         { cout << "A B2 was just constructed " << endl; }
      ~B2() { cout << "A B2 will be destroyed with " << data.get() << endl; }
      B2& operator=(const B2& b2)
                   { cout<< "I am performing a B2 assignment" << endl;
                     data = b2.data; return *this;}
      B2 process() { data.inc(); return *this; }
      void print() { cout << "The B2 data is: " << data.get() << endl ; } };
```

```
class B3 { A* data;
    public:
      B3(const A& a){
         data = new A(a);
         cout <<  "A B3 was just constructed " << endl;}
      B3(const B3& b3) {
          data = new A(*(b3.data));
          cout << "A B3 was just constructed by copying " << endl; }
      ~B3() { cout << "A B3 will be destroyed " << endl;
              delete data; }
      B3& operator=(const B3& b3)
            { cout<< "I am performing a B3 assignment" << endl;
              delete data; data = new A(*(b3.data)); return *this; }
      B3 process() { data->inc(); return *this; }
      void print() { cout << "The B3 data is: " << data->get() << endl ; } };

class C { B1 data1; B2 data2; B3 data3;
   public:
     C(const B1& b1, const B2& b2, const B3& b3) :
       data1(b1), data2(b2), data3(b3)
       { cout << "A C was just constructed " << endl; }
     ~C() { cout << "A C will be destroyed! " << endl; }
     B1 process1() { return data1.process();}
     B2 process2() { return data2.process();}
     B3 process3() { return data3.process();} };

int main()
    { A a; B1 b1(a); B2 b2(a); B3 b3(a); C c(b1,b2,b3);
      a.print(); b1.print(); b2.print(); b3.print();
      b1 = c.process1(); b2 = c.process2();
      c.process2(); b3 = c.process3();
      cout << "////////////////////////////" << endl;
      a.print(); b1.print(); b2.print(); b3.print(); }
```

3. Προκειμένου να ολοκληρώσετε τις σπουδές σας, ενδέχεται να εκπονήσετε Πτυχιακή Εργασία. Η εργασία αυτή θα εντάσσεται σε μια επιστημονική περιοχή των σπουδών σας. Κατά την εκπόνηση της εργασίας, θα πρέπει να μελετήσετε και τη σχετική βιβλιογραφία. Όταν ολοκληρωθεί η εργασία σας, θα πρέπει να συντάξετε και να παραδώστε μια αναφορά (κείμενο) στην οποία να την περιγράφετε. Στην αναφορά αυτή θα πρέπει να συμπεριλάβετε και την βιβλιογραφία ("references") την οποία μελετήσατε με τη μορφή συνόλου "βιβλιογραφικών αναφορών".

   Μια βιβλιογραφική αναφορά ("reference") μπορεί να είναι είτε κάποιο άρθρο από επιστημονικό περιοδικό ("article") είτε εργασία σε πρακτικά συνεδρίου ("inproceedings") είτε κεφάλαιο από βιβλίο ("inbook") είτε βιβλίο ("book") είτε διατριβή ("thesis") είτε σελίδα του παγκόσμιου ιστού ("url").

   Μια βιβλιογραφική αναφορά χαρακτηρίζεται από τρεις λέξεις-κλειδιά που αποτελούν χαρακτηριστικά της περιοχής που εντάσσεται (π.χ. προγραμματισμός, λειτουργικά συστήματα, αλγόριθμοι).

   Ένα άρθρο από επιστημονικό περιοδικό συμπεριλαμβάνει τα ονόματα των συγγραφέων, τον τίτλο του άρθρου, τον αριθμό του τόμου του περιοδικού, έναν επιπλέον αριθμό που αφορά υποδιαίρεση του τόμου, τον εκδότη του περιοδικού (π.χ. Elsevier) και χρονολογία έκδοσης του άρθρου.

   Μια εργασία σε πρακτικά συνεδρίου συμπεριλαμβάνει τα ονόματα των συγγραφέων, τον τίτλο της εργασίας, τον τίτλο του συνεδρίου, τον εκδότη των πρακτικών και χρονολογία διεξαγωγής του συνεδρίου.

Ένα κεφάλαιο από βιβλίο συμπεριλαμβάνει τα ονόματα των συγγραφέων, τον τίτλο του κεφαλαίου, τον τίτλο του βιβλίου, τον εκδότη του βιβλίου και τη χρονολογία έκδοσης του βιβλίου.

Ένα βιβλίο συμπεριλαμβάνει τα ονόματα των συγγραφέων, τον τίτλο του βιβλίου, τον εκδότη του βιβλίου και τη χρονολογία έκδοσης του βιβλίου.

Μια διατριβή συμπεριλαμβάνει το όνομα του συγγραφέα, τον τίτλο της, το όνομα του ιδρύματος στο οποίο εκπονήθηκε (π.χ. Ε.Κ.Π.Α.) και τη χρονολογία της.

Μια σελίδα παγκόσμιου ιστού συμπεριλαμβάνει τη διεύθυνσή της και την ημερομηνία της τελευταίας επίσκεψής σας σε αυτήν.

Κάθε μία από τις παραπάνω αναφορές αρχικοποιείται με τα στοιχεία που συμπεριλαμβάνει. Δίδεται η δυνατότητα συνολικής εκτύπωσης των στοιχείων της (print) καθώς και η δυνατότητα ανάκτησης των λέξεων-κλειδιών που τη χαρακτηρίζει (get_keywords).

Για στατιστικούς λόγους, μας ενδιαφέρει να μπορούμε να μάθουμε, πόσες αναφορές από το κάθε είδος έχει η βιβλιογραφία μας (get_no_article), (get_no_inproceedings), (get_no_inbook), (get_no_book), (get_no_thesis) και (get_no_url).

Η βιβλιογραφία μιας εργασίας αρχικά είναι κενή. Δίδεται η δυνατότητα προσθήκης μιας αναφοράς στη βιβλιογραφία (add_reference). Στην περίπτωση αυτή, προστίθεται μια ήδη υπάρχουσα αναφορά στη βιβλιογραφία. Μπορεί να γίνει εκτύπωση όλων των αναφορών μιας βιβλιογραφίας (print). Επίσης, δίδεται η δυνατότητα εκτύπωσης των στατιστικών της (print_stats). Τέλος, δίδεται η δυνατότητα εξαγωγής των τριών κυριαρχουσών λέξεων-κλειδιών των αναφορών της βιβλιογραφίας (get_keywords).

Υλοποιήστε την παραπάνω πληροφορία σε C++, υλοποιώντας κατάλληλες κλάσεις. (*Σημείωση:* Δεν χρειάζεται υλοποίηση συνάρτησης main).