Αντικειμενοστραφής Προγραμματισμός

Χρήση αντικειμένων

Συνέχεια

Αντικείμενα ως ορίσματα συναρτήσεων - 1

- ✓ Στο πρόγραμμα που ακολουθεί ορίζεται μια συνάρτηση μέλος addBalance(), η οποία έχει ως στόχο να προσθέσει τα ποσά δύο διαφορετικών λογαριασμών.

```
#include <iostream.h>
class Account
{
   private:
         float  balance;
   public:
         Account()          // συνάρτηση κατασκευής χωρίς ορίσματα
         {
            balance = 0;
         }
         Account(float  balance1)    // συνάρτηση κατασκευής με όρισμα
         {
            balance = balance1;
         }
```

Αντικείμενα ως ορίσματα συναρτήσεων - 2

```
void  withdraw(float  money)
{
   if (money <= balance)
      balance = balance - money;
   else
      cout << "Το ποσό ανάληψης υπερβαίνει το τρέχον!" << endl;
}
void deposit(float  money)
{
   balance += money;
}
float getBalance()
{
   return  balance;
}
void addBalance(Account  x, Account y)
{
   balance = x.balance + y.balance;
}
};
```

Αντικείμενα ως ορίσματα συναρτήσεων - 3

```
main()
{
   Account  ac1(100.0), ac2(70.0), ac3;

   ac3.addBalance(ac1, ac2);
   cout << "Τρέχον ποσό λογαριασμού ac1:" <<ac1.getBalance() << endl;
   cout << "Τρέχον ποσό λογαριασμού ac2:" << ac2.getBalance() << endl;
   cout << "Συνολικό ποσό λογαριασμών:" << ac3.getBalance() << endl;
}
```

- Η κλίση της συνάρτησης στο κυρίως πρόγραμμα:

ac3.addBalance(ac1,ac2);

Μεταβιβάζει τα αντικείμενα ac1 και ac2 στις παραμέτρους της συνάρτησης x και y αντίστοιχα.

- Στον κώδικα της συνάρτησης γίνεται αναφορά στα δεδομένα των αντικειμένων x.balance και y.balance. Αυτό είναι εφικτό γιατί, αν και τα δεδομένα είναι ιδιωτικά, είναι προσπελάσιμα καθώς η συνάρτηση είναι μέλος της κλάσης.
- Τα δεδομένα προστίθενται και το αποτέλεσμα αποδίδεται στο δεδομένο του αντικειμένου το οποίο καλεί την συνάρτηση-μέλος, το οποίο είναι το ac3.

Επιστροφή αντικειμένων από συναρτήσεις

Στο προηγούμενο παράδειγμα είδαμε πως τα αντικείμενα μεταβιβάζονται ως ορίσματα σε συναρτήσεις. Στο παράδειγμα που ακολουθεί θα δούμε μια συνάρτηση που επιστρέφει ένα αντικείμενο.

```
Account addBalance(Account ac)
{
   Account temp;
   temp.balance = balance + ac.balance;
   return temp;
}
main()
{
   Account ac1(100.0), ac2(70.0), ac3;
   ac3 = ac1.addBalance(ac2);
   cout << "Τρέχον ποσό λογαριασμού ac1:" << ac1.getBalance() << endl;
   cout << "Τρέχον ποσό λογαριασμού ac2:" << ac2.getBalance() << endl;
   cout << "Συνολικό ποσό λογαριασμών:" << ac3.getBalance() << endl;
 }
```

Επιστροφή αντικειμένων από συναρτήσεις

- Το παραπάνω παράδειγμα πετυχαίνει ότι ακριβώς και το προηγούμενο, δηλαδή προσθέτει δύο λογαριασμούς και το συνολικό ποσό αποδίδεται σε ένα τρίτο αντικείμενο, μόνο που χρησιμοποιεί μια διαφορετική προσέγγιση.

- Στη συνάρτηση addBalance() μεταβιβάζεται μόνο το ένα αντικείμενο ως όρισμα, π.χ το ac2 μεταβιβάζεται στην παράμετρο ac.

Ένταξη κλάσης στους τύπους δεδομένων

- Μια κλάση είναι ένας πλήρης τύπος δεδομένων, όπως λόγου χάρη οι τύποι int και double με διευρυμένες βέβαια λειτουργίες.

- Μπορούμε να έχουμε:
  i. Μεταβλητές τύπου κλάσης (αντικείμενα)
  ii. Παραμέτρους συναρτήσεων τύπου κλάσης
  iii. Επιστρεφόμενες τιμές συναρτήσεων τύπου κλάσης

Πίνακες

Πίνακας

- Ένας πίνακας είναι μία ομάδα δεδομένων του ιδίου τύπου. Οι πίνακες δίνουν την δυνατότητα επεξεργασίας πολλών δεδομένων με γρήγορο και από τρόπο.

  Πχ. Για την επεξεργασία των βαθμών των σπουδαστών σε ένα τμήμα απαιτείται μεγάλο πλήθος μεταβλητών. Για 50 σπουδαστές χρειαζόμαστε 50 μεταβλητές (v1,v2, v3…v50). Πράξεις και γενικά χειρισμός τόσων μεταβλητών είναι πρακτικά αδύνατος.

- Με την χρήση πίνακα όλες οι μεταβλητές έχουν ένα κοινό όνομα και διαφέρουν μόνο ως προς τον δείκτη.

Ορισμός-Δήλωση πίνακα

- Η δήλωση-ορισμός γίνεται ως εξής:

  type name[size]

  Π.χ. float vathmos[50]

  // dilonei ena pinaka me 50 theseis kai stoixeia pragmatikous

  Π.χ. int v[5]

  // dilonei ena pinaka me 5 theseis kai stoixeia akeraious

- Η προσπέλαση των στοιχείων γίνεται ως εξής
  - Το 1ο στοιχεία του ανωτέρω πίνακα είναι το v[0],
  - Το 2ο στοιχείο είναι το v[1]
  - Το τελευταίο στοιχείο του πίνακα είναι το v[4] στη περίπτωση που έχει δηλωθεί ένας πίνακας v[5] ή το v[49] στην περίπτωση που έχει δηλωθεί ένας πίνακας v[50]

Αρχικοποίηση πίνακα

- Τιμές σε πίνακα μπορούν να δοθούν με τον παρακάτω τρόπο
  int v[5]={101,102,103,104,105 }
- Αν το μέγεθος του πίνακα δεν έχει γραφεί τότε με μία αρχικοποίηση το πλήθος των στοιχείων του πίνακα ορίζεται από το πλήθος των τιμών τις τιμές που πίνακα
  int v[ ]={ 100,200,300}
  Με τον τρόπο αυτό ορίζεται πίνακας 3 θέσεων
- Αν στην αρχικοποίηση ενός πίνακα παραληφθούν κάποιες τιμές τότε είναι 0
  int v[5]={101,102}
  Οι υπόλοιπες θέσεις του πίνακα δηλ. v[2] =0 , v[3] =0 , v[4] =0
- Η δήλωση int v[50]={0} μηδενίζει όλες τις θέσεις σε ένα πίνακα

Δήλωση πίνακα

Δήλωση πίνακα

int a[10];

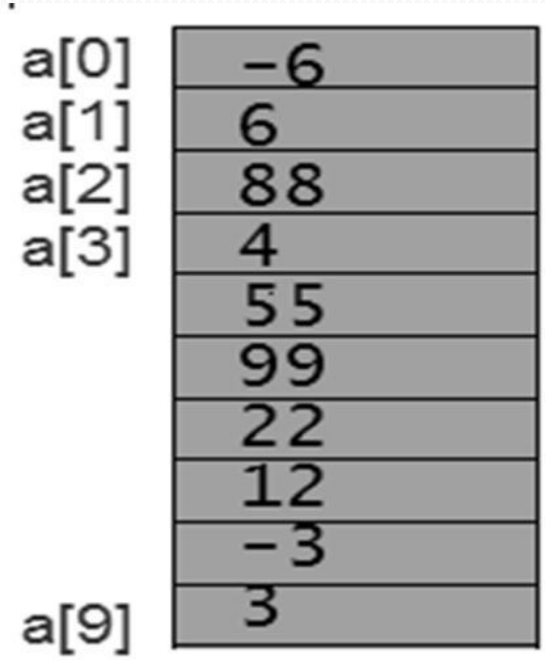
Όνομα πίνακα a

Τύπος int

Σύνολο στοιχείων 10

Συνολικά bytes

4X10=40



Μέγεθος πίνακα-Συνάρτηση sizeof

- Η συνάρτηση sizeof μπορεί να χρησιμοποιηθεί με πίνακες και επιστρέφει σαν τιμή το συνολικό μέγεθος σε bytes που καταλαμβάνει ο πίνακας.

```
cout<<sizeof(int);          //4
cout<<sizeof(double);       //8
int i; sizeof(i) ;          //4
```

- Όλα τα στοιχεία του πίνακα έχουν το ίδιος μέγεθος. Για να βρεθεί το μέγεθος που καταλαμβάνει ένας πίνακας μπορεί να χρησιμοποιηθεί η εντολής

```
int plithos = sizeof(pin)/sizeof(pin[0])
```

- Το ίδιο θα ήταν αν χρησιμοποιηθεί η εντολή

```
float pin[ ];
int plithos = sizeof(pin)/sizeof(float)
```

Επεξεργασία πίνακα

- Γίνεται με την χρήση της εντολής for.
- Ο λόγος που χρησιμοποιείται το for, αντί για κάποια άλλη δομή επανάληψης, είναι γιατί είναι γνωστό το πλήθος των επαναλήψεων που πρέπει να γίνουν για την πρόσβαση σε όλα τα στοιχεία του πίνακα.

Εισαγωγή και εμφάνιση στοιχείων (1)

```
int iArr[10];          //Array of size 10 is defined
int i,j;
cout<<"Input the elements of the array:"<<endl;
                       //As last subscript is 9
for(i=0; i<10; i++) {

   cin>>iArr[i];       //i is called the array subscript
}
cout<<"The elements stored in the array are:"<<endl;
                       //Displays the elements of the array
for(j=0; j<10; j++)
   cout<<iArr[j]<<" ";
```

Εισαγωγή και εμφάνιση στοιχείων (2)

```
int iArr[10];
int i,j;
cout<<"Input the elements of the array:"<<endl;

for(i=0; i<10; i++) {
   cout<<"Dose to "<< i+1<<" stoixeio ";
   cin>>iArr[i];
}
cout<<"The elements stored in the array are:"<<endl;

for(j=0; j<10; j++)
   cout<<iArr[j]<<" ";


system("PAUSE");
return EXIT_SUCCESS;
```

Εισαγωγή και εμφάνιση στοιχείων (3)

```
Input the elements of the array:
Dose to 1 stoixeio 33
Dose to 2 stoixeio 22
Dose to 3 stoixeio 55
Dose to 4 stoixeio -90
Dose to 5 stoixeio -20
Dose to 6 stoixeio -55
Dose to 7 stoixeio 45
Dose to 8 stoixeio 35
Dose to 9 stoixeio 67
Dose to 10 stoixeio 76
The elements stored in the array are:
33 22 55 -90 -20 -55 45 35 67 76

Press any key to continue . . .
```

Πολυδιάστατος πίνακας

- Ένα πολυδιάστατος πίνακας έχει περισσότερες από μία διαστάσεις. Πίνακες με παραπάνω από 3 διαστάσεις συναντιόνται σπάνια στην πράξη.

Δήλωση πίνακα δύο διαστάσεων

- Ένα πίνακας με δύο διαστάσεις που θα είχε την δυνατότητα να αποθηκεύει τους βαθμούς μίας τάξης 50 ατόμων σε 6 μαθήματα θα οριζόταν ως εξής

```
float vat[50][6];
```

- Για τους πίνακες δύο διαστάσεων ισχύουν ότι και για τους μονοδιάστατους πίνακες σχετικά με την αρχικοποίηση και τον χειρισμό τους

Πίνακας δύο διαστάσεων

- Αρχικοποίηση

  int pinax[2][2] = { { 11, 22 }, { 33, 44 } };

- Προσοχή στην ομαδοποίηση

  int pinax[2][2] = { { 11 }, { 3, 4 } };

- Αναπαράσταση πίνακα δύο διαστάσεων

  int pin[3][4]

	Στήλη 0	Στήλη 1	Στήλη 2	Στήλη 3
Γραμμή 0	pin[0] [0]	pin[0] [1]	pin[0] [2]	pin[0] [3]
Γραμμή 1	pin[1] [0]	pin[1] [1]	pin[1] [2]	pin[1] [3]
Γραμμή 2	pin[2] [0]	pin[2] [1]	pin[2] [2]	pin[2] [3]

Παράδειγμα 1

Να εμφανίζεται πίνακας δύο διαστάσεων στην οθόνη.

Ο πίνακας να αρχικοποιείται μέσα στο πρόγραμμα.

```
const int NROW = 3;
const int NCOL = 5;

int main()
{
      int k, j;
      // ta stoixeia tou pinaka
       int arr[NROW][NCOL]

          ={10, 20, 30, 40, 50 ,
            11, 21, 31, 41, 51 ,
            12, 21, 32, 42, 52 };

      // emfanisi tou pinaka
       for (k = 0; k < NROW; k++) {
          for (j = 0; j < NCOL; j++)
             cout << setw(4)<<arr[k][j] << " ";
          cout << endl;
       }
       cout << endl << endl << endl;
```

```
  10   20   30   40   50
  11   21   31   41   51
  12   21   32   42   52
```

Παράδειγμα 2

```
int i,j;
float a[5][2];

for (i=0; i<5; i++)
{
   for (j=0; j<2; j++)
   {
      cout<<"\n Dose to (" <<i+1<<" ," <<j+1;
      cout<<" ) stoixeio tou pinaka ";
      cin>>a[i][j];
   }
}
```

```
Dose to (1 ,1 ) stoixeio tou pinaka 11
Dose to (1 ,2 ) stoixeio tou pinaka 22
Dose to (2 ,1 ) stoixeio tou pinaka 33
Dose to (2 ,2 ) stoixeio tou pinaka 44
Dose to (3 ,1 ) stoixeio tou pinaka 55
Dose to (3 ,2 ) stoixeio tou pinaka 66
Dose to (4 ,1 ) stoixeio tou pinaka 77
Dose to (4 ,2 ) stoixeio tou pinaka 88
Dose to (5 ,1 ) stoixeio tou pinaka 99
Dose to (5 ,2 ) stoixeio tou pinaka 1111
```

Άσκηση 1: Εμφάνιση μεγαλύτερου στοιχείου

Να εισάγονται σε πίνακα 6 αριθμοί. Μετά την εμφάνιση του πίνακα να εμφανίζεται το μεγαλύτερο στοιχείο του πίνακα και η θέση του.

```
const int LIM = 6;

int main()
{
   int table[LIM];
   int i, maxpos;
   int max;
   cout<<"Iha ginei eisagogi "<<LIM<<" arithmon \n\n\n";
   for (i = 0; i < LIM; i++)  {
      cout<<"Dose ton "<<i+1<<" arithmo : ";
      cin>>table[i];
   }
   cout<<" Ta stoixeia tou pinaka einai"<<endl;
   for (i = 0; i < LIM; i++)
      cout<<table[i]<<" ";
```

Άσκηση 1: Λύση

```
maxpos = 0;
max = table[0];

for (i = 0; i < LIM; i++)
    if (table[i] > max) {
       max = table[i];
     maxpos = i;
    }

cout << endl<<endl;
cout <<"Megistos  =  "<<max ;
cout << ",  stin thesi = "<<maxpos<<endl;
```

Άσκηση 2: Εμφάνιση βαθμολογίας

```
cout << showpoint << fixed << right<<setprecision(2);
float theo[15]={6,3,8,2,7,4,9,5,7,4,10,3,7,8,9};
float erga[15]={3,2,7,8,9,6,5,5,5,8,7,10,5,7,3};
float math[15];
cout<<" a/a Theoria Ergastirio    Sxolio          vathmos "<<endl<<endl;
for (int i=0;i<15;i++)
{
   cout<<setw(3)<<i+1<<" "<<setw(8)<<theo[i]<<" "<<setw(8)<<erga[i]<<" ";
   if (theo[i]<5 && erga[i]<5)
      cout<<"    Apotixia          "<<"------"<<endl;
   else if (theo[i]<5 )
      cout<<"    Apot. theoria     "<<"------"<<endl;
   else if (erga[i]<5)
      cout<<"    Apot. ergastirio "<<"------"<<endl;
   else {
      math[i]=(theo[i]+erga[i])/2;
      cout<<"    Epitixos          ";
      cout<<math[i]<<endl; }
   }
```

Παρατήρηση

- showpoint: Όταν εμφανίζονται αριθμοί εμφανίζονται πάντα με δεκαδικό σημείο
- fixed: Οι αριθμοί εμφανίζονται με τόσα δεκαδικά ψηφία όσα καθορίζονται από την setprecision
- right: Στοιχίζει δεξιά
- setprecision(2): καθορίζει το πλήθος των δεκαδικών ψηφίων

Άσκηση 2: Εμφάνιση βαθμολογίας

a/a	Theoria	Ergastirio	Sxolio	vathmos
1	6.00	3.00	Apot. ergastirio	------
2	3.00	2.00	Apotixia	------
3	8.00	7.00	Epitixos	7.50
4	2.00	8.00	Apot. theoria	------
5	7.00	9.00	Epitixos	8.00
6	4.00	6.00	Apot. theoria	------
7	9.00	5.00	Epitixos	7.00
8	5.00	5.00	Epitixos	5.00
9	7.00	5.00	Epitixos	6.00
10	4.00	8.00	Apot. theoria	------
11	10.00	7.00	Epitixos	8.50
12	3.00	10.00	Apot. theoria	------
13	7.00	5.00	Epitixos	6.00
14	8.00	7.00	Epitixos	7.50
15	9.00	3.00	Apot. ergastirio	------

Πίνακες ως δεδομένα κλάσεων

Πίνακες ως Δεδομένα Κλάσεων

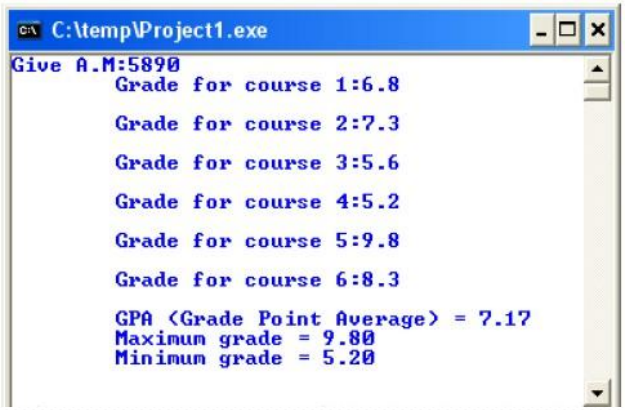
- ✓ Οι πίνακες μπορούν να χρησιμοποιηθούν ως στοιχεία δεδομένων στις κλάσεις.

- ✓ Στο παρακάτω παράδειγμα, ορίζεται μια κλάση με το όνομα Student και μεταβλητές τον αριθμό μητρώου ενός σπουδαστή και ένα πίνακα, που αποθηκεύονται 6 βαθμολογίες του για έξι αντίστοιχα μαθήματα.

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
const int N = 6;
class Student
{
private:
        int  am;
        float  vath[N];
public:
        Student();
        void  readData();
        float  getAvg();
        float  getMin();
        float  getMax();
};
```

```
Student :: Student()
{
        int i;
        am = 0;
        for (i=0; i<N; i++)          vath[i] = 0;
}
void Student :: readData()
{
        int i;
        cout << "Give A.M:";        cin >> am;
        for (i=0; i<N; i++)
        {
                cout << "\tGrade for course " << i+1 << ":";
                cin >> vath[i];    cout << endl;
        }
 }
float Student :: getAvg()
{
        int i;
        float  sum = 0, avg;
        for (i=0; i<N; i++)   sum += vath[i];
        avg = sum/N;
        return avg;
}
float Student :: getMin()
{
        int i;
        float minim;
        minim = vath[0];
        for (i=1; i<N; i++)  if (vath[i] < minim)  minim = vath[i];
   return minim;
 }
```

```
float Student :: getMax()
{
        int i;
        float  maxim;
        maxim = vath[0];
        for (i=1; i<N; i++)
                if (vath[i] > maxim)  maxim = vath[i];
        return maxim;
 }
```




```
main()
{
        Student  s1;
        s1.readData();
        cout << "\tGPA (Grade Point Average) = “
                << fixed << setprecision(2) << s1.getAvg() << endl;
        cout << "\tMaximum grade = “
                << setprecision(2) << s1.getMax() << endl;
        cout << "\tMinimum grade = “
                << setprecision(2) << s1.getMin() << endl;

        cout << endl << endl;
        system("PAUSE");
}
```

Πίνακες αντικειμένων

- ✓ Όπως ένα αντικείμενο μπορεί να περιέχει ένα πίνακα, μπορεί να ισχύει και το αντίστροφο, δηλαδή να έχουμε ένα πίνακα που να περιέχει αντικείμενα.

- ✓ Στο παρακάτω παράδειγμα, ορίζεται μια κλάση με το όνομα Employee και μεταβλητές το όνομα ενός υπαλλήλου, τον μισθό του και τον κωδικό του. Θα πρέπει να διαβάζουμε τον αριθμό των υπαλλήλων μιας επιχείρησης και να εμφανίζουμε τα στοιχεία όλων των υπαλλήλων της.

```
#include <cstdlib>
#include <iostream>

using namespace std;

const int N = 4;
class Employee
{
private:
        int rn;
        char name[20];
        float salary;
```

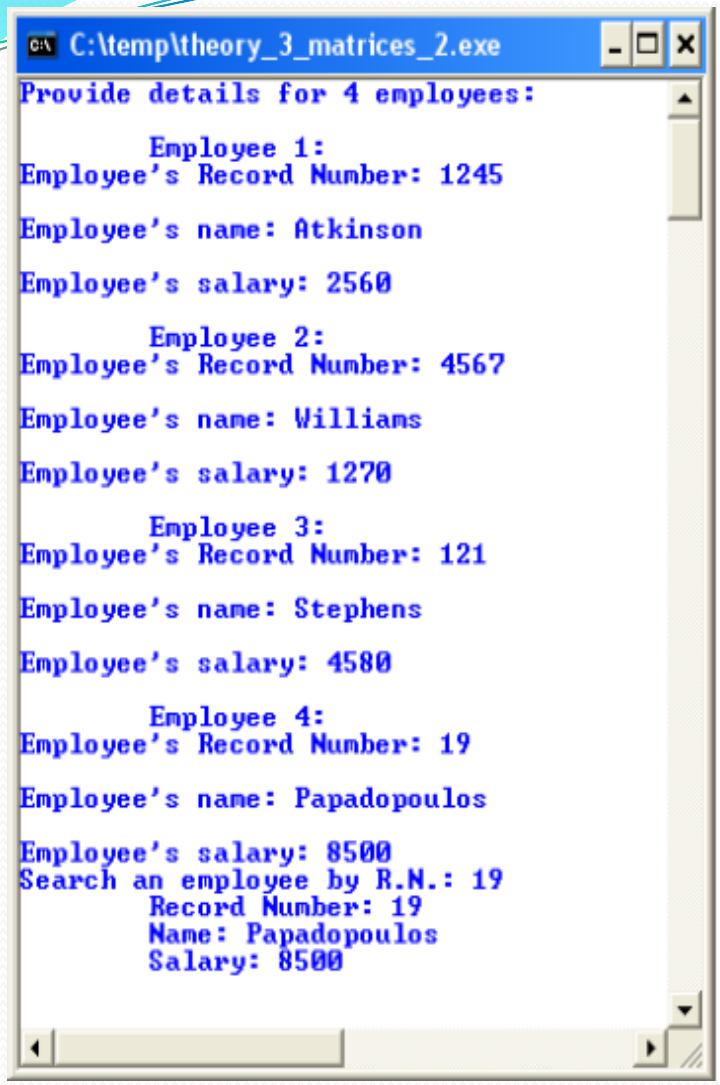
```
public:
  void readData();
  void printData();
  int getRN();
};          // τέλος της κλάσης
```

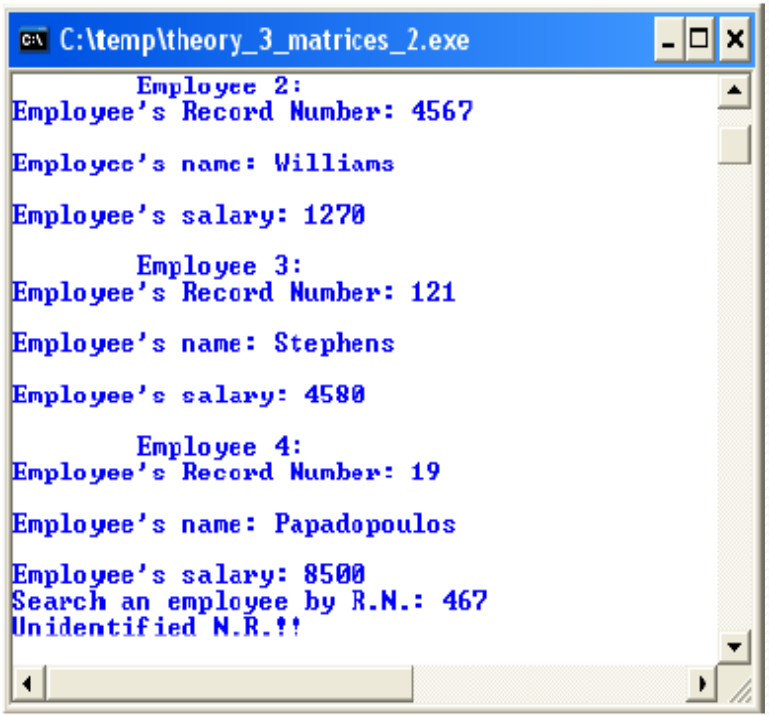
```
void Employee :: readData()
{
        cout << "Employee's Record Number: ";
        cin >> rn;
        cout << endl << "Employee's name: ";
        cin >> name;
        cout << endl << "Employee's salary: ";
        cin >> salary;
}
void Employee :: printData()
{
        cout << "\tRecord Number: " << rn << endl;
        cout << "\tName: " << name << endl;
        cout << "\tSalary: " << salary << endl;
}

int Employee :: getRN()
{
        return rn;
```

```
main()
{
 Employee emp[N];
 int  i, armit;
 bool  found = false;
 cout << "Provide details for " << N << " employees:" << endl;
 for (i=0; i<N; i++)
 {
        cout << endl << "\tEmployee " << i+1 << ":" << endl;
        emp[i].readData();
 }
            cout << "Search an employee by R.N.: ";
            cin >> armit;
            i = 0;
            found = false;
            while  (i<N  &&  found == false)
            {
                    if (emp[i].getRN() == armit)          found = true;
                    else       i++;
             }
             if (found == true)          emp[i].printData();
             else                cout << "Unidentified N.R.!!" << endl;

            cout << endl << endl;
            system("PAUSE");
}           // τέλος της main
```


C:\temp\theory_3_matrices_2.exe
Provide details for 4 employees:
Employee 1:
Employee's Record Number: 1245
Employee's name: Atkinson
Employee's salary: 2560
Employee 2:
Employee's Record Number: 4567
Employee's name: Williams
Employee's salary: 1270
Employee 3:
Employee's Record Number: 121
Employee's name: Stephens
Employee's salary: 4580
Employee 4:
Employee's Record Number: 19
Employee's name: Papadopoulos
Employee's salary: 8500
Search an employee by R.N.: 19
Record Number: 19
Name: Papadopoulos
Salary: 8500


C:\temp\theory_3_matrices_2.exe
Employee 2:
Employee's Record Number: 4567
Employee's name: Williams
Employee's salary: 1270
Employee 3:
Employee's Record Number: 121
Employee's name: Stephens
Employee's salary: 4580
Employee 4:
Employee's Record Number: 19
Employee's name: Papadopoulos
Employee's salary: 8500
Search an employee by R.N.: 467
Unidentified N.R.!!

Πίνακες αντικειμένων

- Στο προηγούμενο πρόγραμμα ορίζεται ένας πίνακας αντικειμένων emp τύπου Employee.
- Κατόπιν, με μια δομή επανάληψης for πληκτρολογούνται πληροφορίες για καθένα από τα 4 αντικείμενα του πίνακα καλώντας τη συνάρτηση readData.
- Τέλος πληκτρολογείται ένας αριθμός μητρώου και αναζητείται στον πίνακα αντικειμένων. Εάν υπάρχει, εμφανίζονται οι πληροφορίες για τον συγκεκριμένο εργαζόμενο.

Εφαρμογές

Να δημιουργήσετε μια κλάση κύκλου, η οποία θα αποτελείται:

1.από μια μεταβλητή-μέλος (ιδιότητα) που θα αντιπροσωπεύει την ακτίνα του (κινητής υποδιαστολής διπλής ακρίβειας)

2. από έναν constructor χωρίς παραμέτρους, ο οποίος αρχικοποιεί την ακτίνα με την τιμή 1,

3. έναν destructor, που να μην κάνει τίποτα

4. από μια συνάρτηση-μέλος (μέθοδο) που να μπορεί να αλλάξει την τιμή της ακτίνας και να ονομάζεται set_aktina (θα δέχεται σαν "είσοδο" έναν κινητής υποδιαστολής διπλής ακρίβειας αριθμό, ο οποίος θα αποτελεί την τιμή της ακτίνας)

5.από μια συνάρτηση-μέλος (μέθοδο), χωρίς παραμέτρους, που να επιστρέφει το εμβαδόν του κύκλου (κινητής υποδιαστολής διπλής ακρίβειας) και να ονομάζεται get_embado.

Να γράψετε ένα πρόγραμμα που:

Α. να δηλώνει έναν πίνακα (διάταξη) τεσσάρων αντικειμένων τύπου κύκλου

Β. κατόπιν, επαναληπτικά:

i. να ζητά 4 φορές από τον χρήστη την τιμή της ακτίνας ισάριθμων κύκλων

ii. και μέσω της συνάρτησης-μέλους της κλάσης κύκλου να "περνά" αυτή την τιμή σε καθένα από τα 4 αντικείμενα.

Γ. Στη συνέχεια, να υπολογίζει επαναληπτικά και να εμφανίζει το εμβαδόν του κάθε αντικειμένου-κύκλου.

Ψευδοτυαχαίοι αριθμοί

Τυχαίοι αριθμοί-Ψευδοτυχαίοι (1)

- Στη βιβλιοθήκη <cstdlib> της C++ υπάρχουν οι συναρτήσεις rand και srand που χρησιμεύουν στην δημιουργία μιας ακολουθίας ψευδο-τυχαίων αριθμών.

- Η συνάρτηση rand() επιστρέφει ένα ακέραιο αριθμό στο διάστημα [0,RAND_MAX]. Η τιμή RAND_MAX εξαρτάται από τον compiler. Συνήθως είναι 32767.
- Η συνάρτηση κάνει επιλογή από μία σειρά αριθμών που παράγονται από μαθηματικούς τύπους αλλά δείχνουν τυχαίοι και γι αυτό τον λόγο και ονομάζονται ψευδοτυχαίοι.

Τυχαίοι αριθμοί-Ψευδοτυχαίοι (2)

- Η rand κάνει χρήση μίας τέτοιας ακολουθίας αριθμών και πάντα αρχίζει από το ίδιο σημείο έναρξης παραγωγής των αριθμών αυτών.
- Εάν το πρόγραμμα τρέξει πάλι από την αρχή θα γίνει παραγωγή των ίδιων αριθμών.
- Για τον λόγο αυτό χρησιμοποιείται η συνάρτηση srand(time(0)) η οποία αρχικοποιεί την ακολουθία που παράγει η συνάρτηση rand με διαφορετική τιμή, λαμβάνοντας μία τιμή από την ένδειξη του ρολογιού του υπολογιστή. Η συνάρτηση time βρίσκεται στην βιβλιοθήκη <ctime>) .

Τυχαίοι αριθμοί-Ψευδοτυχαίοι (3)

- Αν χρειαζόμαστε τυχαίους αριθμούς, ακέραιους, σε κάποιο διάστημα από high έως low τότε (low<high) αντί για το διάστημα 0-RAND_MAX, χρησιμοποιείται του ακέραιο υπόλοιπο του τυχαίο αριθμού σύμφωνα με το τύπο
  x=low+rand()%(high-low+1)
  εμφανίζεται ακέραιος αριθμός στο διάστημα [low,low+high] (με w<RAND_MAX)
- πχ. Για να γίνει παραγωγή τυχαίων αριθμών από το 10 έως και το 14
  rand()%5 +10
- πχ. Για προσομοίωση μίας ρίψης ζαριού
  rand()%6 +1

Παράδειγμα

Να εμφανίζονται τα αποτελέσματα 10 ρίψεων που κάνει παίκτης με δύο ζάρια:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
using namespace std;

int main ()
{   int zari1,zari2;

    srand ( time(NULL) );       // initialize random seed
    for (int i=0;i<10;i++){
        zari1 = rand() % 6 + 1; //tixaios apo 1 eow 6: */
        zari2 = rand() % 6 + 1;
        cout<<zari1<<" "<<zari2<<" ";
        if (zari1==4 && zari2==4) cout<<" dortia";
        if (zari1==6 && zari2==6) cout<<" exares";
        cout<<endl;
    }
  system("PAUSE"); return 0;
}
```