1. **Recapitulare operatori**

Un student este caracterizat de un nume si o serie. Un curs are un nume si un numar oarecare de studenti (se utilizeaza si o structura de date care sa retina studentii). Implementati aceste clase, astfel incat, demo-ul de mai jos sa functioneze:

int main() {

curs C1, C2,C3;

cin>>C1;

cout<<C1;

student s;

cin>>s;

C1=C1+s; // se adauga un nou student la curs

cout<<C1;

cin>>C2;

C3=C1+C2; // daca cele 2 cursuri au aceeasi denumire, se “concateneaza” listele de studenti, // altfel nu se face nimic (se returneaza C1).

cout<<C3;

if(C1==C2) // verifica daca 2 cursuri sunt identice ca denumire si lista de studenti

cout<<"Cursuri identice";

else cout<<"Cursuri diferite";

return 0;

}

1. Se considera o clasa oarecare, cu cativa parametri la alegere.  Completati clasele cu metode si supraincarcari de operatori pentru a permite in functia main() urmatoarele operatii:

/\* adaugati un constructor parametrizat care, prin intermediul unui parametru bool sa atribuie valori campurilor clasei de la consola ("true") si din fisier ("false")\*/

Clasa\_voastra ob1, ob2(1 parametru), ob3(2 parametri), ob4 = ob3, ob5 (false);

cin>>ob1;

cout<<ob1<<”  “<< - ob1;     /\* operatorul: -(minus) - schimba semnul campurilor intregi;

++ob1;

ob1++;

int y = ob1;

cout<<y<<endl;

Clasa\_voastra v[3];

v[0] = ob1;

v[1] = v[2] = ob2 – ob3;

if (v[0] != v[2])  v[0] += v[2];

else if ( !v[1] ) cout<<”v[1] nu e zero”;