

Problema 1:

Scrieti clasele necesare astfel încît codul următor sa functioneze corect:

```
int main()
{
    Studenti noteMatematica = { Student{ "Popescu", 10 }, Student{ "Ionescu", 9 }, Student{ "Georgescu", 8 } };
    Studenti noteFizica = { Student{ "Popescu", 7 }, Student{ "Ionescu", 6 }, Student{ "Marin", 4 } };
    Studenti mediiMatematicaFizica = noteMatematica + noteFizica;
    cout << "Georgescu: [Matematica = " << noteMatematica["Georgescu"] << "]" << endl;
    cout << "Ionescu: [Fizica = " << noteFizica["Ionescu"] << "]" << endl;
    for (int tr = 0; tr < mediiMatematicaFizica.GetCount(); tr++)
    {
        cout << "Student: " << mediiMatematicaFizica(tr) << " Medie: " << mediiMatematicaFizica[tr] << endl;
    }
    return 0;
}
```

Operatorul de adunare (+) între doua obiecte de tipul Studenti creeaza un obiect nou care contine toți studentii existenti. Acolo unde un student are doua note, se pastreaza media acestora. De asemenea, pornim de la premiza ca nu exista 2 studenti cu același nume.

Dacă programul rulează corect, va afișa pe ecran următoarele:

```
Georgescu: [Matematica = 8]
Ionescu: [Fizica = 6]
Student:Georgescu Medie: 8
Student:Ionescu Medie: 7
Student:Marin Medie: 4
Student:Popescu Medie: 8
```

Punctajul se va acorda in felul urmator:

Task	Numar puncte
Clasa(clasele) scrie in fisiere .h si .cpp distincte	2 pct
Codul cu main-ul dat ruleaza si afiseaza ce trebuie pe ecran	3 pct
Codul compileaza corect	2 pct
Implementarea constructorilor	6 pct
Implementarea funcției GetCount	2 pct
Implementarea operatorului de adunare +	6 pct
Implementarea operatorilor de indexare []	6 pct
Implementarea operatorului pentru apel funcție ()	3 pct
Total	30 pct