

## Proiect 1 Programare Orientata Obiect:

Proiectul 1 consta in alegerea a minim 4 clase care sa exemplifice o aplicatie, exemple: magazine de jocuri video, Netflix, etc.

Pentru toate clasele din proiecte, trebuie sa se respecte conceptul de incapsulare a datelor, altfel proiectul se noteaza cu 0.

Proiectele care nu ruleaza sunt notate cu 0.

*30.10.2023*

Proiectul trebuie trimis pana la **finalul saptamanii 4, adica 06.03.2022 ora 23:59** si prezentat in **saptamana 5, in cadrul laboratorului**.

Cine nu este present la prezentare nu beneficiaza de corectatul proiectului si este notat cu 0.

Proiectul trebuie sa contine comentarii.

Proiectele sunt trimise pe adresa de mail [poolaborator@gmail.com](mailto:poolaborator@gmail.com) si se trimit doar **main.cpp** **redenumit** astfel: **Grupa\_Nume\_Prenume\_Proiect1.cpp**, orice alta denumire sau alt fisier nu este luat in considerare pentru corectare.(Nu se accepta header files)

Pentru orice tentativa de frauda proiectul este notat cu 0.

Proiectul trebuie sa se realizeze in limbajul de programare C++.

Fiecare tema aleasa pentru proiect trebuie sa fie unica la nivel de semigura.

Trebuie sa va alegeti tema si sa o notati in excel-ul aferent (link excel:

~~<https://docs.google.com/spreadsheets/d/1DZNK2LOQR-YprAcfq5h2xS5utLewKOIPzVD8ZGIKKYU/edit?usp=sharing>~~ -

## Cerinte :

1. Construiti-vă o mini aplicație care gestionează o afacere/agendă/planificare etc. 0.25
2. Aplicația trebuie să conțină minim 4 clase. 0.25
3. Alegerea claselor trebuie să fie cat mai logică, astfel incat fiecare clasă sa aibă legătură una cu cealaltă. 0.25
4. Fiecare clasă trebuie să contină minim 4 variabile, iar în tot proiectul să se regasească toate tipurile de variabile de mai jos:

- - Int/long 0.2
- - Bool 0.2
- - Char 0.2
- - Char\* ~~0.3~~ 0.5
- - Floate 0.2
- - Int\*/Floate\* (SAU) 0.5
- ~~○ - Double~~
- - Const ~~0.3~~ 0.5
- - Static 0.3

Se pot adăuga și alte variabile în funcție de necesitate.

5. Pentru 2 din clase să se construiască Setters și Getters, iar în tot proiectul să existe Setters și Getters pentru variabilele listate mai sus. (Se pot adăuga Setters și Getters la nevoie)

6. În fiecare clasa trebuie să existe:

- Constructorul cu toți parametrii 0.3
- Cel puțin 2 constructori cu parametrii 0.3 < 0.15
- Constructorul fără parametrii 0.3
- Constructorul de copiere 0.3
- Destructor 0.3
- Forma suprainsarcata a operatorului = 0.5

- Supraincarcarea operatorului << și >> 0,5

7. Sa se supraincarce ~~pentru fiecare clasa~~:

- Operatorul de indexare [] 0,3
- Operatorul ++ sau -- (cele două forme) 0,3
- Minim ~~2~~ operatori matematici (+,-,\* sau /) (sa se respecte comutativitatea) 0,5
- ~~○ Operatorul cast explicit sau implicit~~
- Un operator condițional (<, >=, >=) 0,3
- Operatorul pentru egalitate (==) 0,3

1P

8. Fiecare clasa sa contine o funcționalitate. (De exemplu, daca avem o clasa Agenda, să existe o funcție care îmi afișează datele în care sunt liber in luna August)

1P

9. Pentru 2 din clasele alese sa existe operatori matematici care sa permita operatiile intre cele doua clase.

Exemplu: Pentru clasa A si clasa B supraincarcam operatorul + astfel incat sa fie permise urmatoarele: A+B, A+A+B, A+B+A+B....

1P

10. Proiectul trebuie sa fie insotit de un meniu interactiv care ne poate permite cel putin operatiile de baza (CreateReadUpdateDelete) dar si sa poata sa exemplifice functionalitatile aplicatiei. (Putem adauga cate optiuni dorim)

Succes! 😊