

Laborator 3

Instructiuni generale

Acceptati invitatia pe GitHub Classroom pentru laboratorul 3, link:

- <https://classroom.github.com/a/opEG5cqC>

Dupa accept, clonati repository-ul pe calculator:

- `git clone <link_repository>`

Creati fisierele necesare rezolvării laboratorului.

Toata logica programului trebuie implementata in fisierul `.cpp`, cu prototipurile in fisierul header `.h`.

In `main.cpp` scrieti doar apelurile si testarea functiilor.

La final, urcati modificarile:

- `git add .`
 - `git commit -m "Laborator 3"`
 - `git push`
-

Fisierele trebuie impartite astfel:

- **student.h** → declararea clasei
- **student.cpp** → implementarea clasei
- **main.cpp** → testarea functiilor

Fisier: Student.h

```
class Student {
    char *nume;
    int varsta;
    float nota;
public:
    Student(); // constructor fara parametri
    Student(const char *n, int v, float nt); // constructor cu parametri
    ~Student(); // destructor
    void afisare(); // afiseaza attributele
    void modif(char *nume, int varsta, float nota); // modifica attributele clasei
    void modif(const Student &obj); // modifica attributele clasei pe baza unei instante
    float getNota() const; // returneaza nota
    const char* getNume() const; // returneaza numele
};
```

Cerinte (in main):

- 4p – creati un obiect si testati toate metodele in main
- 2p – creati un vector alocat dinamic de tip Student
- 1p – sortati vectorul crescator dupa nota
- 1p – afisati studentul cu cea mai mare nota
- 1p – eliberati memoria ocupata de vector

1 punct din oficiu

Cum se compileaza si ruleaza

1. Compileaza programul:
 - g++ -g main.cpp student.cpp -o lab3
2. Ruleaza simplu:
 - ./lab3
3. Ruleaza cu Valgrind:
 - valgrind ./lab3

Exemple de mesaje:

- “definitely lost” → memorie ne-eliberata
 - “Invalid read/write” → acces invalid dupa eliberare sau in afara vectorului
-