Obiettivi: classi: costruttori e composizione di classi

- 1) Definire la classe esame nel file **esame.h** contenente le seguenti componenti:
 - metodi
 - o costruttore di default esame()
 - o costruttore esame(string): imposta solo il nome dell'esame
 - o costruttore esame(string,int): imposta nome e voto ottenuto dell'esame
 - o getEsame: restituisce il nome dell'esame
 - o setEsame(string nome): imposta il nome dell'esame
 - o getVoto: restituisce il voto ottenuto per l'esame se sostenuto
 - o setVoto: imposta il voto ottenuto per l'esame
 - o isSostenuto: restituisce true se l'esame è stato sostenuto
 - campi di esemplare
 - Nome esame: stringaSostenuto: booleanVoto esame: intero

esame.h

```
#ifndef _ESAME_H
#define _ESAME_H

#include

class esame{
public:
    esame();
    esame(string);
    esame(string, int);
    // ... da completare ...
private:
    string nome;
    // ... da completare ...
};

#endif // _ESAME_H
```

2) Implementare la classe esame nel file esame.cpp esame.cpp

```
#include
esame::esame(){}
// ... da completare ...
```

- 3) Testare la classe esame utilizando un tester **TestEsame.cpp**
- 4) Definire la classe studente nel file **studente.h** e implementarla nel file **studente.cpp**: (Si suppone che uno studente debba fare al massimo 5 esami)
 - metodi:
 - o costruttore di default studente(): imposta i nomi di default dei 5 esami e i rispettivi flag sostenuto a false
 - o costruttore studente(nome1, nome2, nome3, nome4, nome5): imposta in nomi dei 5 esami usando i parametri nome1, ... e i rispettivi flag sostenuto a false
 - o setMatricola, setNome, getMatricola, getNome
 - o getNumEsami: restituisce 5 (numero massimo di esami da sostenere)
 - o setEsame(Voto,i): imposta il voto dell'esame i-esimo
 - getVotoEsame(i), getNomeEsame(i), getSostenuto(i)
 - o getMedia(): restituisce la media degli esami sostenuti
 - o printStudente(): visualizza tutte le informazioni dello studente compresi gli esami sostenuti e

1 di 2 30/10/2020, 11:07

i rispettivi voti

- campi di esemplare
 - o matricola: string
 - o nome: string

esami: vettore di 5 esamiconst numesami: intero

- 5) Testare la classe studente utilizando un tester Studente.cpp
- 6) Definire la classe studenti avente come campo di esemplare un vettore di 3 studenti e implementare alcuni metodi di set, get e print.

2 di 2