ID: 2101013

Problem:

A Student Management System in a University with classes called Student, Alumni and Freshers.
The Student class will have attributes such as ID, Name, CGPA and methods such as grade and display.
The Alumni will publicly inherit from the Student class to reuse the attributes such as name and id and others.
The Freshers will publicly inherit from the Alumni class to reuse the attributes such as session and others.
The display() function is made virtual so that if it were to be accessed via pointer by any derived class, it can access its own overridden function and not the base class function.
There is also a friend function to give access to change the department of a student.

Code:

```
#include<iostream>
#include<string>
using namespace std;


class Student{
	protected:
		int id;
		string name;
		float cgpa;

	public:
		Student(int ID, string NAME, float CGPA){
			id=ID;
			name=NAME;
			cgpa=CGPA;
		}

		string grade(){
			if(cgpa>0 && cgpa<=2)
				return "FAIL";
			else if(cgpa<=3)
				return "C";
			else if(cgpa<=3.5)
				return "B";
			else if(cgpa<=4)
				return "A";
		}
```

```
		virtual void display(){
			cout<<"Name: "<<name<<endl;
			cout<<"ID: "<<id<<endl;
			cout<<"Grade: "<<grade()<<" ("<<cgpa<<")"<<endl;

		}
};

class Alumni:public Student{
	protected:
		string grade;
		string session;

	public:
		Alumni(int id, string name, float cgpa, string Grade, string
Session):Student(id, name, cgpa){
			grade=Grade;
			session = Session;
		}
		void display(){
			cout<<"Name: "<<name<<endl;
			cout<<"ID: "<<id<<endl;
			cout<<"Session: "<<session<<endl;
			cout<<"Grade: "<<grade<<" ("<<cgpa<<")"<<endl;

		}
};

class Freshers:public Student{
	float gstMark;
	string department;
	string session;
	public:
		Freshers(int id, string name,float cgpa, string Session, float GstMark,
string Dpt):Student(id, name, cgpa){
			gstMark=GstMark;
			department=Dpt;
			session=Session;
		}
		void display(){
			cout<<"Name: "<<name<<endl;
			cout<<"ID: "<<id<<endl;
			cout<<"Dept.: "<<department<<endl;
			cout<<"Session: "<<session<<endl;
			cout<<"GST mark: "<<gstMark<<endl;

		}
		friend void migrate(Freshers& f);
};
```

```
void migrate(Freshers& f){
	cout<<"Enter new Dept.: ";
	cin>>f.department;
}

int main(){
	Student s1(21013, "MehrFarz", 3);
	s1.display();

	Alumni a1(20012, "AliAtei",4, "A", "2018-19");
	a1.display();

	Freshers f1(22012, "HenaHarith",0, "2023-24", 150.56, "IRE");
	f1.display();

	migrate(f1);
	f1.display();
	return 0;
}
```

Output:

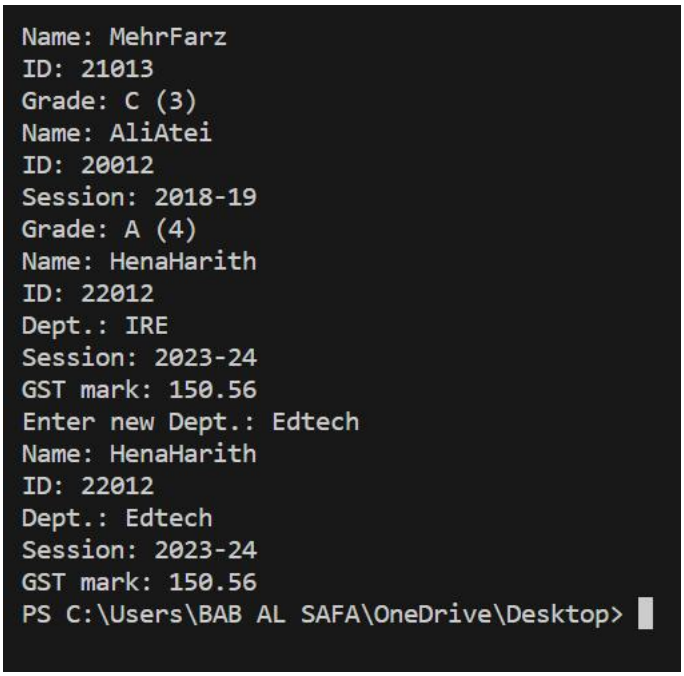

Fig 1: Output on Console.