

American International University-Bangladesh

Faculty of Science and Information Technology

Department of Computer Science

OBJECT ORIENTED PROGRAMMING 1 (JAVA)

|  |
| --- |
| **Semester: Summer 2019-20** |

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Sec** |
| **19-41289-3** | **Saha, Anik Kumar** | **AA** |
| **19-41283-3** | **Ahmed, Nafiz** | **AA** |

1. **Project Title:** Student Management
2. **Project Description:** This project is based on Student Management. There is a class for account creation and login. After logging in, you will be provided with 4 options. Among them you can select any one to do your required operation. And you will be continuously asked to select any option until you press exit. After pressing exit at last when everything will be done, all your information will be stored in a text file which is done in file class from where you can read your given inputs at any time you want.
3. **Project Features:** Features of this project are given below,
4. It can create account for admin.
5. After that it allows you to login.
6. Here you can add students up to 50.
7. Each student contains student’s Name, ID, Department and CGPA.
8. You can also remove students by their ID.
9. After adding and removing, there is another feature that allows you to see information of remaining students.
10. And last but not the least, information of remaining students can be stored in a text file.

**Project Demonstration**

**Topics that we have covered in this project:**

* Array
* Inheritance
* Encapsulation
* Abstraction
* Polymorphism
* Association
* Exception handling
* File IO
* Interface
* Package

|  |
| --- |
| interface RecordInterface |
| public void removeDetails(String id) ;  public void showDetails() ;  public void checkId(String id) ; |

**Classes**

|  |
| --- |
| interface adminInterface |
| public void newAccount() ;  public void login() ; |

|  |
| --- |
| Admin |
| public String UserName ;  public String Password ;  public void newAccount()  public void login() |

|  |
| --- |
| AllRecord |
| static student info[ ] = new student[ ] ;  public void insertDetails(student a)  public void removeDetails(String id)  public void showDetails()  public void checkId(String id) |

|  |
| --- |
| Operations |
| int input ;  public void task() |

|  |
| --- |
| fileclass |
| fileclass()  public static void writeFile()  public static void Readf() |

HAS

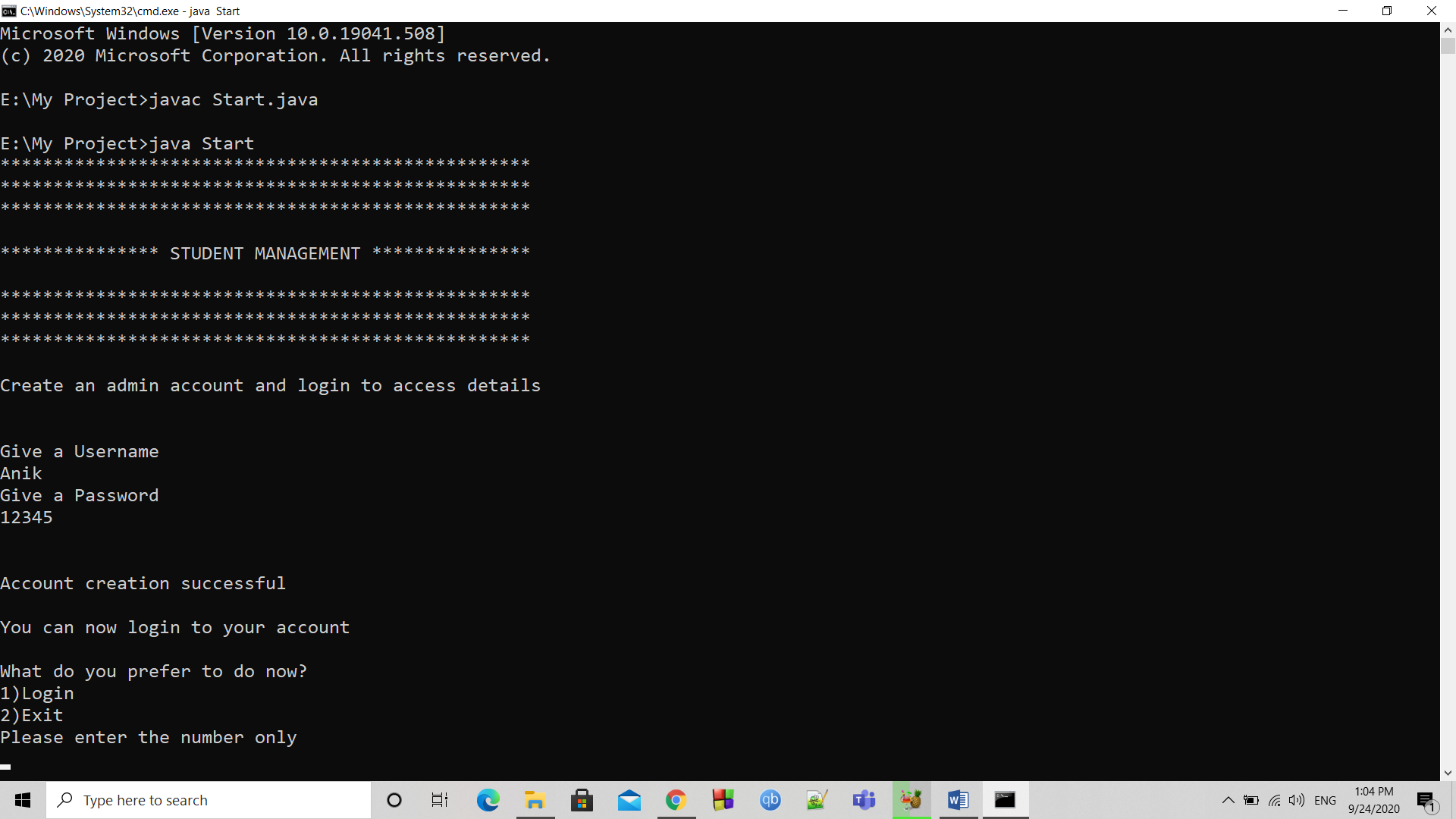
|  |
| --- |
| Student |
| student()  student(String name , String id , String department , String cgpa)  public void setName(String Name)  public void setId(String Id)  public void setDepartment(String Department)  public void setCgpa(String Cgpa)  public String getName()  public String getId()  public String getDepartment()  public String getCgpa() |

|  |
| --- |
| abstract class CommonFeatures |
| protected String Name ;  protected String Id ;  protected String Department ;  protected String Cgpa ;  abstract void setName(String Name) ;  abstract void setId(String Id) ;  abstract void setDepartment(String Department) ;  abstract void setCgpa(String Cgpa) ;  abstract String getName() ;  abstract String getId() ;  abstract String getDepartment() ;  abstract String getCgpa() ; |

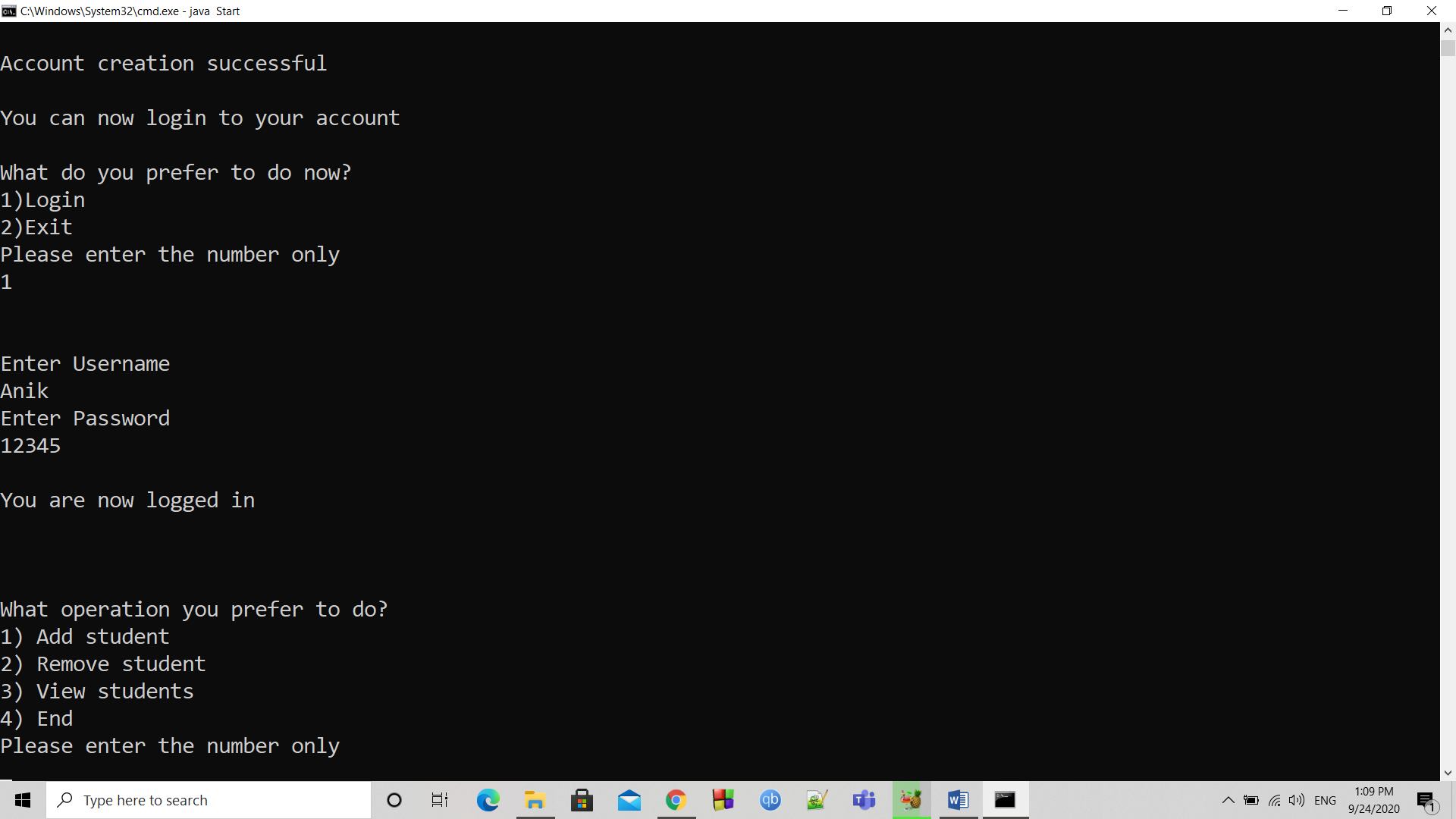
Lastly there is a Start class to demonstrate all the classes and their uses.

**Output of our project**

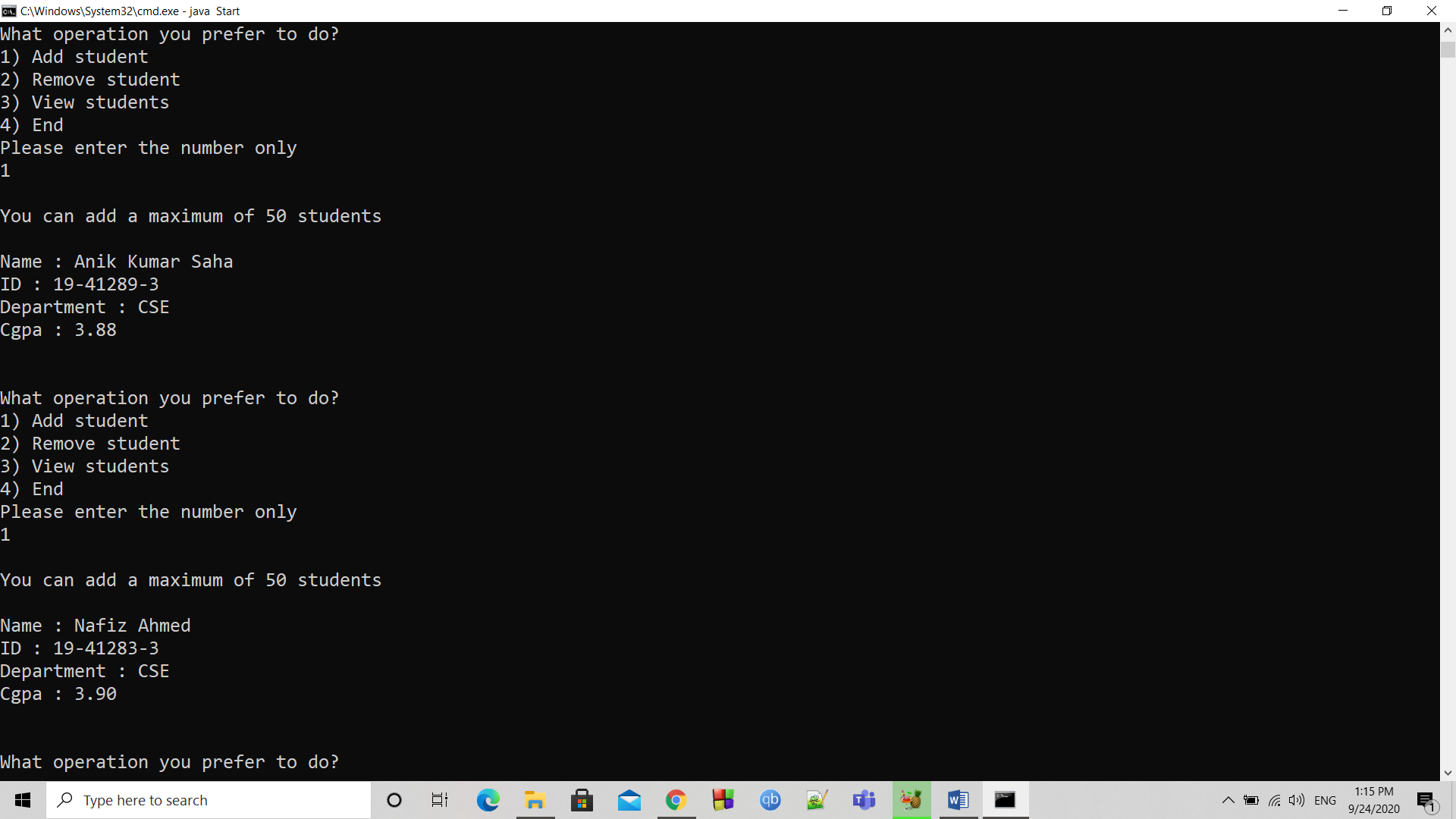
Creating admin account:

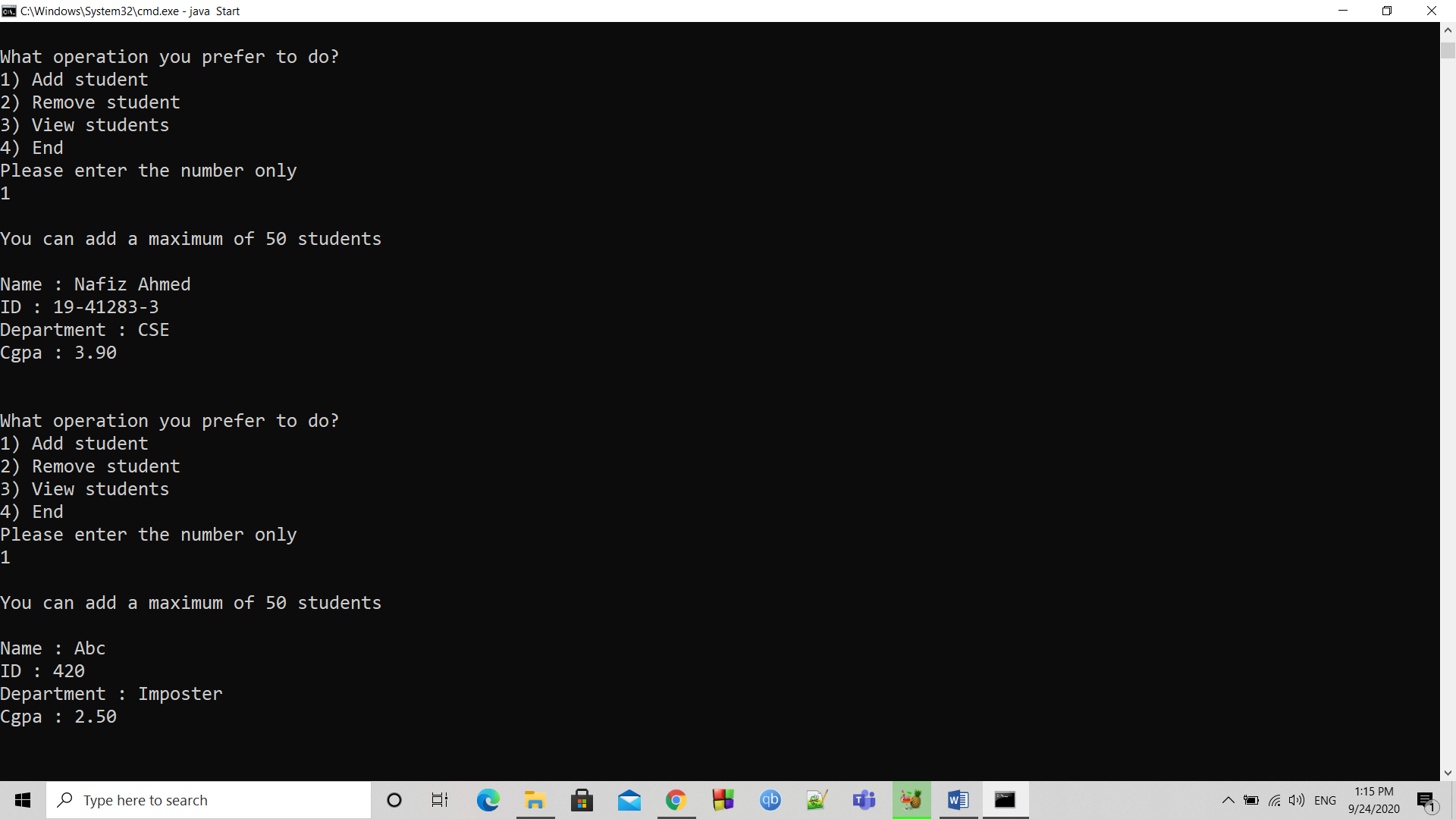


Logging in:

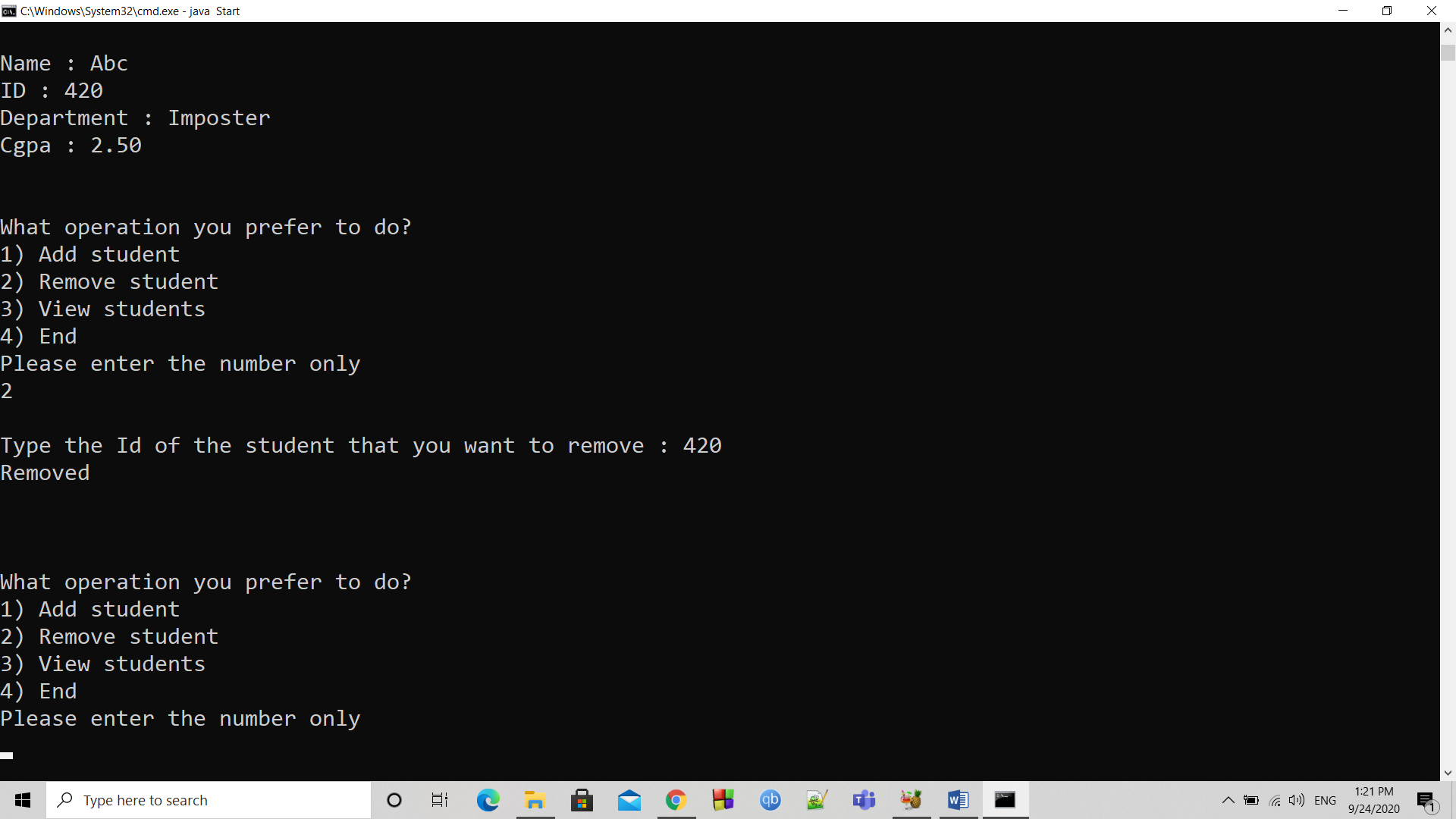


Adding students:

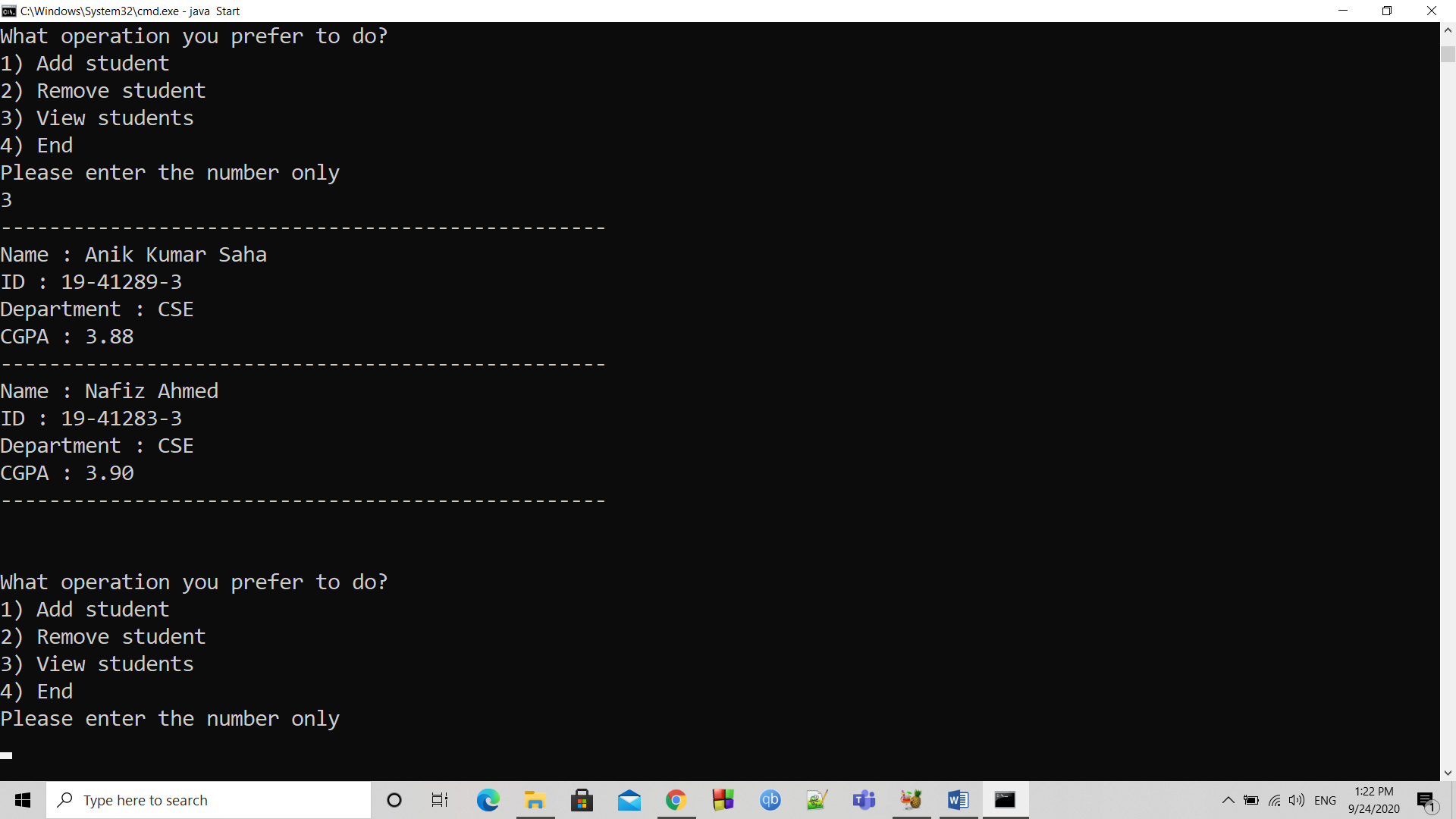




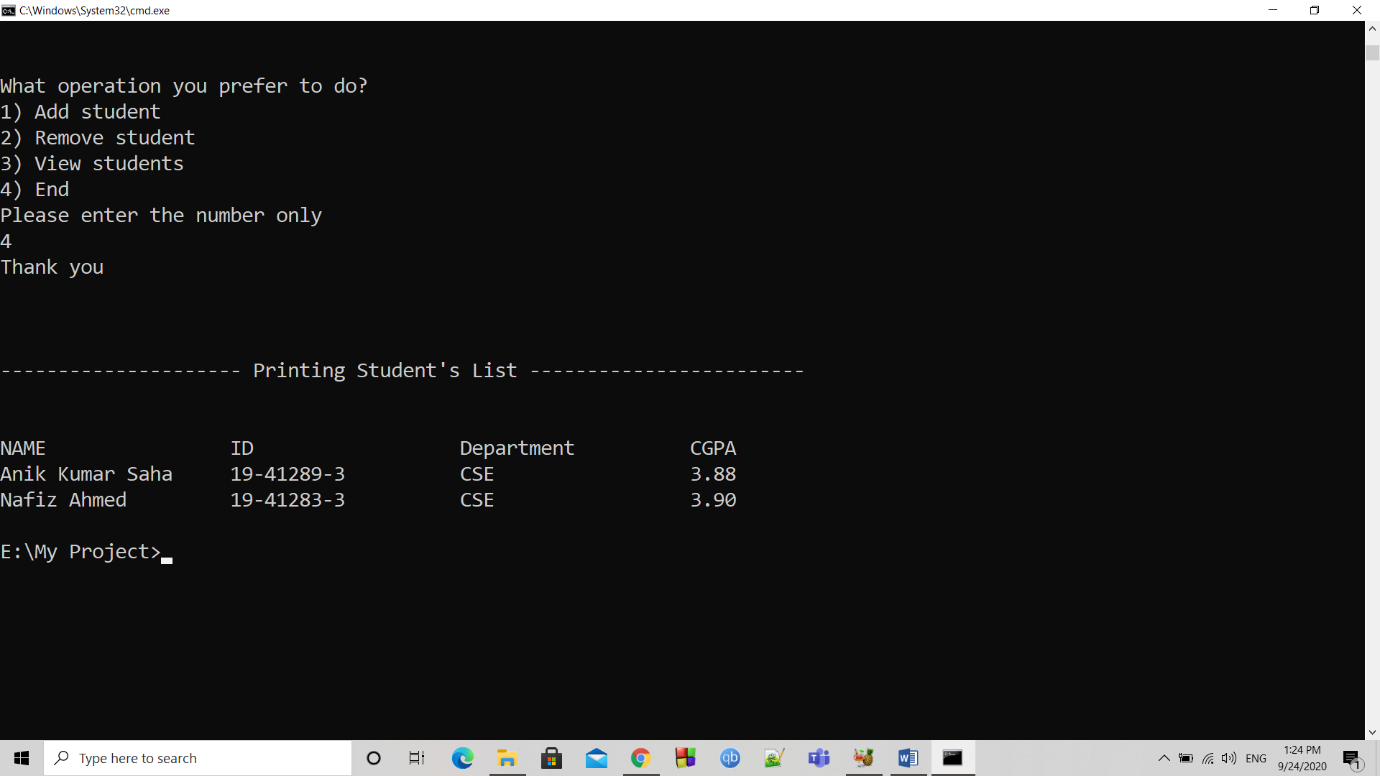
Removing student:



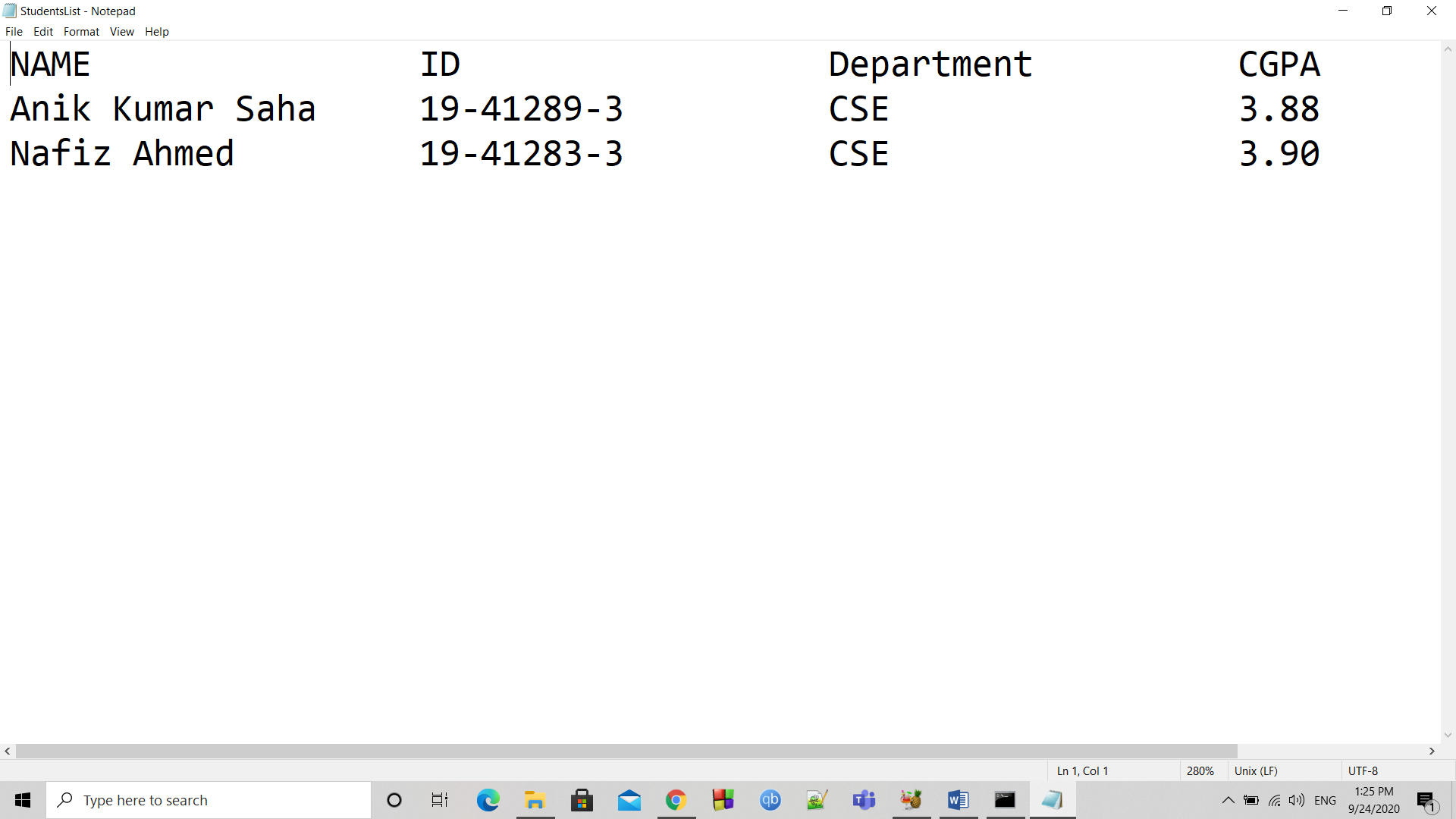
Showing remaining students:



Ending management:



Workings of file IO:



**THE END**