**URL to GitHub Repository:**

**URL to Your Coding Assignment Video:**

**Instructions:**

* In Visual Studio Code, write the code that accomplishes the objectives listed below and ensures that the code compiles and runs as directed.
* Create a new repository on GitHub for this week’s assignments and push this document, with your project code, to the repository.
* Include the URLs for this week’s repository and video where instructed.
* Submit this document as a .PDF file in the LMS.

**Coding Steps:**

* Create a menu app as seen in this week’s video. What you create is up to you as long as it meets the following requirements:
  + Use at least one array.
  + Use at least two classes.
  + Your menu should have the options to **create**, **view**, and **delete** elements.

**Video Steps:**

* Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed.
* This video should be done using screen share and voice over.
* This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend.
  + You can create a new meeting, start screen sharing, and start recording.
  + This will create a video recording on your computer.
* This should then be uploaded to a publicly accessible site, such as YouTube.
  + Ensure the link you share is **PUBLIC** or **UNLISTED**!
  + If it is not accessible by your grader, your project will be graded based on what they can access.

class student {

    constructor(name, major){

        this.name = name;

        this.major = major;

    }

    describe(){

        return `${this.name} studies ${this.major}.`;

    }

}

class university {

    constructor(name){

        this.name = name;

        this.student = [];

    }

    addStudent(student) {

        if (student instanceof student){

            this.students.push(student);

        } else {

            throw new Error (`You can only add an instance of student. Argument is not a student: ${student}`);

        }

}

describe() {

    return `${this.name} has ${this.students.length} students.`;

}

}

class Menu {

    constructor(){

        this.universities = [];

        this.selectedUniversity = null;

    }

    Start () {

        let selection = this.showMainMenuOptions();

        while (selection != 0){

            switch(selection){

                case '1':

                    this.createUniversity();

                    break;

                case '2':

                    this.viewUniversity();

                    break;

                case '3':

                    this.deleteUniversity();

                    break;

                case '4':

                    this.displayUniversities();

                    break;

                    default:

                        selection = 0;

            }

            selection = this.showMainMenuOptions();

        }

        alert ('Goodbye!');

}

showMainMenuOptions() {

    return prompt(`

    0) exit

    1) create new university

    2) view university

    3) delete university

    4) display all universities

    `);

}

    showUniversityMenuOptions(universityInfo){

        return prompt(`

        0) back

        1) create student

        2) delete student

        --------------

        ${universityInfo}

        `);

    }

    displayUniversities(){

        let universitystring = '';

        for (let i = 0; i < this.universities.length; i++){

            universitystring += i + ') ' + this.universities [i].name + `\n`;

        }

        alert(universitystring);

    }

    createUniversity(){

        let name = prompt('enter name for new university:');

        this.universities.push(new university(name));

    }

    viewUniversity(){

        let index = prompt('enter the index of the university you wish to view:');

        if (index > -1 && index < this.universities.length){

            this.selectedUniversity = this.universities[index];

            let description = 'university name : ' + this.selectedUniversity.name + '\n';

            for (let i=0; i< this.selectedUniversity.students.length; i++){

                description += i + ') ' + this.selectedUniversity.students[i].name + ' - ' +

                this.selectedUniversity.students[i].major + '\n';

            }

        let selection = this.showUniversityMenuOptions(description);

        switch (selection){

            case '1':

                this.createStudent();

                break;

            case '2':

                this.deleteStudent();

        }

    }

    }

    deleteUniversity(){

        let index = prompt('enter the index of university you wish to delete:');

        if (index > -1 && index < this.university.length){

            this.university.plice(index, 1);

        }

    }

    createStudent(){

        let name = prompt('enter name for new student:');

        let major = prompt('enter major for new student:');

        this.selectedUniversity.students.push(new students(name, major));

    }

    deletePlayer(){

        let index = prompt('enter the index of the student you wish to delete:');

        if (index > -1 && index < this.selectedUniversity.students.length){

            this.selectedUniversity.student.splice(index, 1);

        }

    }

}

let menu = new Menu();

menu.Start();