

## به نام خدا



## درس برنامه نویسی پیشرفته

### تمرین چهارم

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۲ - ۱۴۰۱

مهلت ارسال :

۱۴۰۲/۰۲/۱۰

مباحث :

استراکچر - پراپرتی

مسئول تمارین :

آریا شهنسوار

## فهرست

۳	<input type="checkbox"/> آداب نامه تمرینات
۴	<input type="checkbox"/> نکات تمرین سری چهارم
۵	<input type="checkbox"/> تمرین ۱ . آنلاین شاپ
۱۰	<input type="checkbox"/> تمرین ۲ . مدیریت دانشگاه

## آداب نامه تمرینات

- پاسخ تمامی سوالات تنها به زبان های C# قابل قبول می باشد .
- علیرغم اعتماد کامل تیم تی ای به شما دانشجویان عزیز، تمامی کد های شما با سایر دانشجویان بصورت خودکار و توسط برنامه مقایسه خواهند شد . همچنین در طول ترم ، از تمامی پاسخ های شما ارائه گرفته خواهد شد و نحوه کار تمامی بخش های هر سوال از شما پرسیده خواهد شد، لذا از کپی نمودن کد دوستانتان خودداری کنید و تمامی پاسخ ها، کد خودتان باشد . همچنین از آنجایی که مشورت و هم فکری با سایر دوستان بسیار کار پسندیده و مفیدی است – برخلاف کپی کردن کد (:-) در صورت هم فکری با دانشجوی دانشجویان ، نام وی را بصورت کامنت شده در ابتدای کد خود بنویسید.
- برای ارسال تمرین در طول ترم، در مجموع ۷ روز می توانید تاخیر داشته باشید و در صورتی که جمع تاخیر دانشجویی بیشتر از ۷ روز شود، تمرین وی قابل قبول نخواهد بود لذا تلاش کنید تمرینات را در زمان مقرر در سامانه آپلود کنید.
- در تمامی تمرینات سعی شده است که سوالات ساده تر در ابتدا و سوالات دشوار تر در انتهای فایل قرار گیرند ( از ساده به دشوار مرتب شده اند ) .
- در صورت وجود هرگونه سوال در مورد تمرینات ، سعی کنید تا جایی که امکان دارد سوال خود را در گروه پرسید چرا که شاید سوال شما، سوال دوستان نیز باشد و دوستانتان نیز بتوانند از پاسخ سوال شما بهره ببرند.

## نکات تمرین سری چهارم

- سوالات را در سامانه کوئرا و در قسمت تمرین چهارم آپلود نمایید .
- در تمامی سوالات باید مدیریت خطا بطور کامل و دقیق صورت گیرد و بخشی از نمره هر سوال به Exception Handling درست و مناسب تعلق میگیرد .
- در این سری از تمرین نیازی به نوشتن Unit test برای هیچ یک از سوالات نیست.
- در پیاده سازی کلاس ها ، دقت کنید که تمام مواردی که در سوال از شما خواسته شده است با دقت پیاده سازی شود.
- از آنجایی که هر سوال توسط یک تی ای طرح شده است ، تنها تی ای طراح آن سوال می تواند شما را بصورت دقیق راهنمایی کند به همین منظور طراح هر سوال در زیر نوشته شده است تا در صورت ابهام و پرسش در مورد هر سوال، در صورتی که نیاز به پرسش سوال بصورت انفرادی در پیوی هست ، به تی ای مربوطه مراجعه بفرمایید.

○ سوال ۱ . خانم کاوسی نژاد ، آقای شهسوار

○ سوال ۲ . خانم عربعلی ، آقای شهسوار

## تمرین ۱. آنلاین شاپ

سارا یک فروشگاه لباس آنلاین دارد و برای ارسال لباس ها به پست نیاز دارد. بنابراین میخواهد برنامه ای برای سامان دهی ارسال ها بنویسد.

### آیتم های مورد نظر و اینام های آن :

#### Clothes:

انواع لباس ها شامل مقادیر: Tshirt = , Shirt = 4 , Skirt = 3 , Pants = 2 , Socks = 1 , Hat = 7 , Gloves = 6 , 5 میباشد .

#### Postline:

انواع پست کردن شامل مقادیر: Certified = 4 , Media = 3 , Priority = 2 , Regular = 1 , Ground = 6 , Airmail = 5 میباشد .

#### Currency:

انواع واحد پول شامل مقادیر: Dollar = 30 , Pound = 35 , Dirham = 7 , Franc = 29 میباشد .

#### Vehicle:

انواع وسایل نقلیه جهت ارسال مرسوله های آنلاین شاپ شامل مقادیر: Motorcycle = 1 , Car = 3 , Truck = 2 میباشد .

### استراکچر های مورد نیاز :

#### Location:

این استراکت شامل کشور، استان، شهر، خیابان ، پلاک است.

**نکته ۱:** این استراکت باید به صورت Record تعریف شود و پراپرتی های این استراکت باید به صورت مثال زیر تعریف شوند :

```
using System;
public record struct Product
{
    public string Name { get; init; }
    public int CategoryId { get; init; }
}
public class Program
{
    public static void Main()
    {
        var product = new Product
        {
            Name = "VideoGame",
            CategoryId = 1
        };
        product.Name = ""; // Error CS8852 Init-only property
        // or indexer 'Product.Name' can only be assigned in an object
        // initializer, or on 'this' or 'base' in an instance constructor
        // or an 'init' accessor.
    }
}
```

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/proposals/csharp-10.0/record-structs>

<https://anthonygiretti.com/2021/08/03/introducing-c-10-record-struct/>

## Time:

این استراکت شامل سال ، ماه، روز، ساعت ، دقیقه است.

## PriceInOtherCountry:

این استراکت شامل یک قیمت (از نوع double) و یک واحد پول می باشد. همچنین بوسیله سربارگذاری عملگر \* (operator overloading) معادل آن را به واحد تومان خروجی دهد. (خروجی از جنس **PriceInToman** میباشد).

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/operator-overloading>

[https://www.tutorialspoint.com/csharp/csharp\\_operator\\_overloading.htm](https://www.tutorialspoint.com/csharp/csharp_operator_overloading.htm)

## PriceInToman:

این استراکت شامل یک فیلد قیمت از نوع double است .

## کلاس های مورد نیاز :

### Post:

این کلاس شامل فیلد های : نوع لباس ، آیدی ارسال ، مبدا (location) ، مقصد (location) ، ساعت تحویل به پست (time) و ساعت رسیدن به مقصد (time) ، وسیله نقلیه (vehicle) ، نوع پست (postline) و قیمت پست (PriceInOtherCountry) است . حتما توجه کنید که ساعت رسیدن به مقصد دیر تر از ساعت ارسال باشد. با استفاده از RegEx هنگام مقدار دهی آیدی ها بررسی کنید که در این مقدار ابتدا دو حرف انگلیسی و سپس یک عدد ۴ رقمی و پس از آن یکی از علامت های \$@# وجود داشته باشد

**مثال :** hw2371@ یک شماره پرواز valid و hw2628 یک شماره پرواز invalid است .

**نکته ۲ :** تمام این اطلاعات باید به صورت property ذخیره شوند .

**نکته ۳ :** برای هر کدام از فیلد های کلاس باید از استراکچری که در پرانتز مربوط به آن نوشته شده است استفاده شود .

## منوی برنامه

برنامه باید دارای یک منو باشد که شامل گزینه های زیر است :

سارا برای دسترسی به منوی برنامه کافیس عبارت Manager را وارد کرده و پس از ورود منوی زیر نمایش داده میشود .

- افزودن ارسال پستی (Add Post)
- مرتب سازی ارسال های پستی (Sort Post)
- ویرایش اطلاعات مرسوله پستی (Change Post Data)
- خروج (Exit)

## توضیح روند برنامه

### افزودن ارسال پستی :

با این دستور نوع لباس، آیدی ارسال، مبدأ، مقصد، ساعت تحویل به پست و ساعت رسیدن به مقصد، وسیله نقلیه، نوع پست و قیمت پست ( به همراه واحد قیمت) بعنوان ورودی داده می شود. این تابع ابتدا بوسیله قیمت پست و واحد یک شی از استراکت **PriceInOtherCountry** میسازد و با استفاده از متد موجود در این استراکت، قیمت پست را به واحد تومن به دست می آورد

حال برای هر یک از فیلدهای ارسال با استفاده از اطلاعاتی که از ورودی گرفته شده یک شی از استراکت و کلاس های مورد نظر ساخته و سپس با استفاده از مفاهیم property آنها را مقداردهی میکند و یک ارسال به لیست ارسال ها اضافه میشود.

در این قسمت با استفاده از سربارگذاری، عملگر == را به گونه ای پیاده سازی کنید که بتوان برابری دو شی از کلاس post با نوع داده یکسان را بررسی کرد. یعنی پس از ساخت شی جدید ارسال با همه ی شی های از پیش ساخته شده چک کند و در صورت یکسان بودن خروجی مناسب نشان دهد.

### مرتب سازی ارسال های پستی :

این دستور شامل ۲ قسمت است :

- **سورت بر اساس قیمت:** باید از سارا یک قیمت پیشنهادی دریافت کند و ارسال هایی که با کمترین تفاوت قیمت است را بالاتر از ارسال هایی با تفاوت قیمت بیشتر، نشان دهد.
- **سورت بر اساس مبدأ و مقصد:** باید از سارا یک مبدأ و مقصد دریافت کند و ارسال هایی که هم مبدأ و هم مقصد یکسانی با ورودی سارا دارند، را نشان دهد.



**نکته ۴:** برای نشان دادن خروجی های این دو قسمت از رنگ بندی های مناسب استفاده کنید تا اطلاعات خوانا تر شوند .

### ویرایش اطلاعات مرسوله پستی :

در این دستور سارا میتواند یکی از فیلدهای ارسال با آیدی مورد نظرش را تغییر دهد. یعنی پس از انتخاب این گزینه باید سارا یکی از گزینه ها را انتخاب کرده سپس اطلاعات جدید را وارد کند و با استفاده از مفاهیم property آن را تغییر دهد.(واضح است که location به دلیل init تغییری نمیکند).

### خروج :

با استفاده از این دستور برنامه به پایان میرسد .

**نکته ۵:** بعد از هر افزودن ارسال مرسوله پستی اطلاعات کلاس پست را در فایل post.json به صورت زیر ذخیره کنید و بعد از ویرایش اطلاعات مرسوله پستی آن را آپدیت کنید .

<https://learn.microsoft.com/en-us/dotnet/standard/serialization>

```
{
  "Clothes": "Hat",
  "Count": 1,
  "ID": "st4321@",
  "Vehicle": "Car",
  "Postline": "Certified",
  "TotalPrice": "10",
  "Currency": "Dollar",
  "Origin":
  {
    "Country": "USA",
    "Province": "NewYork",
    "City": "NewYork",
    "Street": "West",
    "Number": "201"
  },
  "Destination":
  {
    "Country": "USA",
    "Province": "NewYork",
    "City": "NewYork",
    "Street": "6thAve",
    "Number": "27"
  },
  "OriginTime": "2023-4-14-21-15",
  "DestinationTime": "2023-4-17-11-50"
}
```

```
{
  "Clothes": "Socks",
  "Count": 4,
  "ID": "po1234#",
  "Vehicle": "Car",
  "Postline": "Regular",
  "TotalPrice": "40000",
  "Currency": "Toman",
  "Origin":
  {
    "Country": "Iran",
    "Province": "Tehran",
    "City": "Tehran",
    "Street": "Hengam",
    "Number": "2"
  },
  "Destination":
  {
    "Country": "Iran",
    "Province": "Tehran",
    "City": "Tehran",
    "Street": "Azadi",
    "Number": "15"
  },
  "OriginTime": "2023-3-15-12-30",
  "DestinationTime": "2023-3-17-9-20"
}
```

## تمرین ۲. مدیریت دانشگاه

در این سوال قصد داریم که یک برنامه به منظور مدیریت یک دانشگاه بنویسیم.  
آیتم های مورد نظر و اینام های آن :

نوع تحصیل : (Tuition)

انواع تحصیل بر اساس پرداخت یا عدم پرداخت شهریه شامل مقادیر :  $Tuition\_free = 0$  ,  $Tuition\_based = 1$  میباشد .

نوع واحد : (Credit)

انواع واحد بر اساس هزینه جهت انتخاب واحد دانشجوی شبانه شامل مقادیر :  $Theory = 4380$  ,  $General = 2200$  ,  $Lab = 7000$  ,  $Final\_Project = 5620$  ,  $Internship = 4100$  میباشد .

استراکچر مورد نیاز :

**Student:**

این استراکچر باید دارای فیلدهای نام ، نام خانوادگی ، سن ، جنسیت و شماره تماس و شماره دانشجویی و رشته دانشجوی باشد . شماره تماس باید حتما ۱۱ رقم باشد و حتما باید با رقم های "۰۹" شروع شود . این استراکچر دارای ۲ متد به نام های EditProfile است . اولین متد برای تغییر رشته که ورودی آن از نوع int (آیدی رشته مورد نظر) میباشد . دومین متد برای تغییر شماره تماس که ورودی آن از نوع string میباشد . ورودی ها در این ۲ متد درواقع مقدار جدید شماره تماس و رشته هستند و با صدا زدن آنها باید مقادیر آپدیت شوند.

**نکته ۱:** جنسیت باید از نوع enum باشد که شامل مقادیر :  $male = 1$  ,  $female = 2$  میباشد .

**Professor:**

این استراکچر باید دارای فیلدهای نام ، نام خانوادگی ، سن ، جنسیت و شماره تماس و شماره پرسنلی و لیست دروس در حال تدریس باشد . شماره تماس . این استراکچر ۱ متد به نام EditPhoneNumber دارد. این متد برای تغییر شماره تماس است

که وروی آن از نوع string است. ورودی در این متد درواقع مقدار جدید شماره تماس است و با صدا زدن آن باید مقدار شماره تماس آپدیت شود.

### کلاس های مورد نیاز :

#### University :

این کلاس دارای ویژگی های **آیدی**، نام ، لیست دانشجویان مشغول به تحصیل در این دانشگاه ، لیست رشته های ارائه شده در دانشگاه و لیست اساتید مشغول در دانشگاه است .

#### Field :

این کلاس دارای ویژگی های **آیدی**، نام ، تعداد واحدکی، لیست دانشجویان مشغول به تحصیل دراین رشته و لیست دروس ارائه شده در این رشته است . تمامی این ویژگی ها به غیر از لیست دروس ارائه شده باید به صورت property پیاده سازی شوند . **متغیر آیدی Field** باید به گونه ای پیاده سازی شود که در خارج از محیط کلاس نتوان آن را مقداردهی کرد و تنها بتوانیم مقدار آن را بخوانیم و توانایی تغییر آن را نداشته باشیم. همچنین این متغیر باید در تابع سازنده مقداردهی شود.

**نکته ۲ :** برای هر property باید getter, setter مناسب نوشته شود.

**نکته ۳ :** آیدی هر رشته باید منحصر به فرد باشد .

#### Course :

این کلاس دارای ویژگی های **آیدی**، نام ، تعداد واحد , روز و ساعت برگزاری , نوع واحد از جنس enum و استاد و ترم ارائه این درس از جنس Semester و لیستی از دانشجویانی که این درس را اخذ کرده اند(به همراه نمرات اخذ شده توسط آنها در این درس ) ,هزینه تمام شده درس و ظرفیت درس است . تمامی این ویژگی ها به غیر از لیست دانشجویان باید به صورت **property** پیاده سازی شوند . **متغیر آیدی Course** باید به گونه ای پیاده سازی شود که در خارج از محیط کلاس نتوان آن را مقداردهی کرد و تنها بتوانیم مقدار آنرا بخوانیم و توانایی تغییر آن را نداشته باشیم. همچنین این متغیر باید در تابع سازنده مقداردهی شود.

**نکته ۴ :** برای هر property باید getter, setter مناسب نوشته شود.

**نکته ۵ :** آیدی هر رشته باید منحصر به فرد باشد .

**نکته ۶:** هزینه تمام شده برای هر درس با توجه به نوع آن ضریب در تعداد واحد تعیین میشود.

### Semester :

این کلاس دارای ویژگی های آیدی ، لیست دروس ارائه شده در این ترم ، لیست اساتید و لیست دانشجویان است. هر درس داده ای از نوع Course است.

**نکته ۷:** بخش آیدی باید از نوع enum باشد که شامل مقادیر :  $Fall = 1$  ,  $Spring = 2$  ,

$Summer = 3$  میباشد و در تابع سازنده این کلاس متغیرهای آیدی دریافت و مقداردهی میشود .

### منوی اصلی برنامه :

- دسترسی به عنوان ادمین ( Admin )
- دسترسی به عنوان استاد ( Professor )
- دسترسی به عنوان دانشجو ( Student )
- خروج ( Exit )

### توضیح روند برنامه :

این برنامه سه مدل دسترسی خواهد داشت شامل :

۱. دسترسی به عنوان ادمین
۲. دسترسی به عنوان استاد
۳. دسترسی به عنوان دانشجو

- دسترسی Admin : ادمین برای دسترسی به این قسمت کافایت عبارت Admin را وارد کرده و سپس عبارت ADMIN را به عنوان پسورد خود وارد کند. پس از وارد شدن به عنوان ادمین منوی زیر نمایش داده میشود :

## منوی ادمین :

- افزودن رشته (Add Field)
- افزودن دانشجو (Add Student)
- افزودن استاد (Add Professor)
- افزودن درس (Add Course)
- تغییر رشته برای دانشجو (Change Student's Field)

- دسترسی **Student** : هر دانشجو برای دسترسی به این قسمت کفایت عبارت Student را وارد کند سپس شماره دانشجویی ثبت شده درسیستم دانشگاه را وارد کند. پس از وارد شدن به عنوان دانشجو منوی زیر نمایش داده میشود :

## منوی دانشجو :

- اخذ درس (Add Course)
- حذف درس (Remove Course)
- محاسبه شهریه (Calculate Tuition)
- نمایش نمره در درس مد نظر (Show Score)
- تغییر شماره تماس (Edit PhoneNumber)

- دسترسی **Professor** : هر استاد برای دسترسی به این قسمت کفایت عبارت Professor را وارد کند سپس شماره پرسنلی ثبت شده درسیستم دانشگاه را وارد کند. پس از وارد شدن به عنوان استاد منوی زیر نمایش داده میشود :

## منوی استاد :

- افزودن نمره به درس (Assign Score To Course)
- سورت کردن دانشجویان براساس نمره در درس مدنظر (Sort Students)
- پیدا کردن میانگین نمرات در درس مد نظر (Average Score)
- تغییر شمار تماس (Edit PhoneNumber)

## توضیح روند برنامه :

**افزودن رشته :** این متد به عنوان ورودی **آیدی رشته جدید** را دریافت خواهد کرد و در صورتی که رشته در دانشگاه **موجود نبود** اطلاعات تکمیلی رشته را دریافت میکند

**نکته ۸:** در صورتی که این رشته در لیست رشته های دانشگاه وجود داشت پیام خطای مناسب جهت تکراری بودن رشته نمایش داده شود .

**افزودن درس به رشته :** این متد به عنوان ورودی **آیدی درس** و **آیدی رشته** را دریافت میکند و در صورت موجود نبودن آیدی درس در آن رشته اطلاعات تکمیلی درس را دریافت کرده به دروس رشته مدنظر اضافه میکند . همچنین به لیست دروسی که استاد تدریس میکند هم افزوده میشود .

**نکته ۹:** در صورت موجود نبودن **آیدی رشته** شی از درس ساخته نشده و پیام خطای مناسب چاپ شود . در صورت موجود بودن آیدی درس پیام خطای مناسب جهت تکراری بودن درس چاپ شود .

**نکته ۱۰:** در صورت برابری روز و ساعت این درس با روز و ساعت درس دیگر ارائه شده توسط استاد این درس پیام خطای مناسب جهت **تداخل** چاپ شود .

**افزودن دانشجو :** این متد به عنوان ورودی **شماره دانشجويي** را دریافت میکند و در صورت موجود نبودن **شماره دانشجويي** در لیست دانشجویان موجود در کلاس دانشگاه **اطلاعات تکمیلی** دانشجو را دریافت کرده و به لیست دانشجویان موجود در کلاس دانشگاه و لیست دانشجویان موجود در کلاس رشته این دانشجو اضافه میکند .

**نکته ۱۱:** در صورت موجود بودن **شماره دانشجويي** پیام خطای مناسب جهت تکراری بودن چاپ شود . **شماره تماس** باید حتما ۱۱ رقم باشد و حتما باید با رقم های "۰۹" شروع شود . دانشجو میتواند **شماره تماس** نداشته باشد .

**افزودن استاد :** این متد به عنوان ورودی شماره پرسنلی را دریافت میکند و در صورت موجود نبودن این شماره پرسنلی در لیست اساتید موجود در کلاس دانشگاه اطلاعات تکمیلی استاد را دریافت کرده و به لیست اساتید موجود در کلاس دانشگاه اضافه میکند .

**نکته ۱۲ :** در صورت موجود بودن شماره پرسنلی در لیست اساتید دانشگاه پیام خطای مناسب چاپ شود.

**تغییر رشته برای دانشجو :** این متد آیدی دانشجو و آیدی رشته جدید را از ورودی گرفته و دانشجو را از لیست دانشجویان رشته قبلی و از لیست دروسی که اخذ کرده حذف میکند و به لیست دانشجویان رشته جدید اضافه میکند و متد موجود در استراکت نیز فراخوانی میشود .

**نکته ۱۳ :** در صورت موجود نبودن آیدی درس و در صورت موجود نبودن آیدی رشته جدید و در صورت یکسان بودن آیدی رشته قدیمی و جدید پیام خطای مناسب چاپ شود .

**اخذ درس :** این متد به عنوان ورودی آیدی درس را دریافت میکند و به لیست دروس اخذ شده توسط دانشجو اضافه میکند و همچنین دانشجو به لیست دانشجویان موجود در کلاس درس نیز اضافه میشود .

**نکته ۱۴ :** در صورت موجود نبودن آیدی درس پیام خطای مناسب چاپ شود . در صورتی که درس مد نظر در دروس رشته دانشجو قرار ندارد پیام خطای مناسب چاپ شود . در صورت تکمیل بودن ظرفیت درس پیام خطای مناسب جهت ناموفق بودن اخذ درس چاپ شود . در صورت برابری روز و ساعت این درس با روز و ساعت درس دیگر اخذ شده توسط دانشجو پیام خطای مناسب جهت تداخل چاپ شود .

**حذف درس :** این متد به عنوان ورودی آیدی درس را دریافت میکند و این درس را از لیست دروس اخذ شده توسط دانشجو حذف میکند و از ظرفیت پر شده درس یک واحد کم میگردد همچنین دانشجو را از لیست دانشجویان موجود در کلاس درس نیز حذف میکند .

**نکته ۱۵ :** در صورت موجود نبودن آیدی درس پیام خطای مناسب چاپ شود . در صورتی که درس مدنظر در دروس اخذ شده توسط دانشجو قرار ندارد پیام خطای مناسب چاپ شود .

**محاسبه شهریه :** ۲ متد تحت عنوان محاسبه شهریه وجود دارد :

اولی به عنوان ورودی آیدی درس وارد میشود و مبلغ پرداختی برای آن درس نمایش داده میشود .  
دومی ورودی ندارد و شهریه کلی ترم دانشجوها توجه به دروس اخذ شده تا به کنون نمایش داده میشود.  
**نکته ۱۶:** در متد اول در صورتی که دانشجو درس را اخذ نکرده بود پیام خطای مناسب چاپ شود .  
در صورتی که دانشجو شبانه نبود پیام خطای مناسب چاپ شود .

**نمایش نمره :** این متد به عنوان ورودی آیدی درس را گرفته و نمره دانشجو در این درس نمایش داده میشود .

**نکته ۱۷:** در صورتی که نمره دانشجو توسط استاد هنوز وارد نشده باشد , در صورت موجود نبودن آیدی درس و در صورتی که دانشجو درس را اخذ نکرده بود پیام خطای مناسب چاپ شود .

**تغییر شماره تماس (دانشجو) :** این متد به عنوان ورودی یک string به عنوان شماره تماس جدید دریافت میکند و شماره تماس دانشجو را آپدیت میکند

**نکته ۱۸:** شماره تماس باید حتما ۱۱ رقم باشد و حتما باید با رقم های "۰۹" شروع شود.

**افزودن نمره :** این متد به عنوان ورودی آیدی دانشجو , آیدی درس و نمره اخذ شده را گرفته و این نمره را به دانشجو در آن درس تخصیص میدهد .

**نکته ۱۹:** در صورتی که استاد درسی با چنین آیدی ارائه نمیدهد , در صورت موجود نبودن آیدی درس , در صورت موجود نبودن آیدی دانشجو و در صورتی که دانشجو درس با این آیدی را اخذ نکرده باشد پیام خطای مناسب چاپ شود . نمره ثبت شده باید بین صفر و بیست باشد .

**نکته ۲۰:** اگر برای این دانشجو در این درس نمره ای ثبت شده پیامی جهت هشدار به استاد جهت تغییر نمره نمایش داده شود .

**سورت دانشجویان بر اساس نمره :** این متد آیدی درس مورد نظر را گرفته و لیست سورت شده دانشجویان همراه با آیدی و نمراتشان را نمایش میدهد .



**نکته ۲۱:** در صورت موجود نبودن آیدی درس و در صورتی که استاد درسی با چنین آیدی ارائه نمیدهد پیام خطای مناسب چاپ شود.

**نکته ۲۲:** اگر این درس به طور کلی نمره دهی نشده بود و یا تمامی دانشجویان نمره نداشتند پیام خطای مناسب چاپ شود.

**میانگین نمرات:** این متد آیدی درس مورد نظر را گرفته و میانگین نمرات دانشجویان در این درس را نمایش میدهد.

**نکته ۲۳:** در صورت موجود نبودن آیدی درس و در صورتی که استاد درسی با چنین آیدی ارائه نمیدهد پیام خطای مناسب چاپ شود.

**نکته ۲۴:** اگر این درس به طور کلی نمره دهی نشده بود و یا تمامی دانشجویان نمره نداشتند پیام خطای مناسب چاپ شود.

**تغییر شماره تماس (استاد):** این متد به عنوان ورودی یک string به عنوان شماره تماس جدید دریافت میکند و شماره تماس دانشجو را آپدیت میکند

**نکته ۲۵:** شماره تماس باید حتما ۱۱ رقم باشد و حتما باید با رقم های "۰۹" شروع شود.