

آشنایی با دوره، راه اندازی محیط و مرور مفاهیم پایه پایتون

بخش اول: معارفه و معرفی ساختار کارگاه

۱. معرفی کارگاه:
 - نام درس: کارگاه عملی برنامه سازی پیشرفته
 - هدف اصلی: تبدیل مفاهیم تئوری برنامه نویسی شیءگرا (که در کلاس نظری می آموزید) به مهارت عملی و کاربردی با استفاده از زبان پایتون.
 - این کارگاه، محلی برای "کثیف کردن دستها" با کدنویسی است و مشارکت فعال دانشجویان ضروری است.
۲. تشریح نقشه راه (Roadmap): با توجه به اینکه دوره بر برنامه نویسی پیشرفته شیءگرا (OOP) تمرکز دارد و به یک پروژه نهایی ختم می شود، نقشه راه ۱۶ هفته ای ما به شکل زیر خواهد بود:

بخش اول: مبانی و اصول شیءگرایی (هفته ۱ تا ۸)

- هفته ۱: آشنایی با دوره و مرور مفاهیم پایه
 - معرفی ساختار دوره، نحوه ارزیابی و انتخاب زبان پایتون.
 - نصب و راه اندازی محیط توسعه (Python, VS Code).
 - مرور سریع متغیرها، انواع داده و دستورات ورودی/خروجی.
- هفته ۲: ساختارهای داده پایه و کنترل جریان
 - کار با لیستها (Lists)، حلقه های for و while و دستورات شرطی if/else.
- هفته ۳: ساختارهای داده پیشرفته و توابع
 - آشنایی با دیکشنریها (Dictionaries) و تاپلها (Tuples).
 - تکمیل مبحث توابع (آرگومانهای پیش فرض و بازگرداندن چند مقدار).
- هفته ۴: ورود به دنیای شیءگرایی (OOP)
 - آشنایی با مفاهیم کلاس (Class)، شیء (Object)، صفت (Attribute) و متد (Method).
 - نوشتن اولین کلاس و استفاده از سازنده `__init__`.

- **هفته ۵: کپسوله سازی (Encapsulation)**
 - مفهوم پنهان سازی داده با صفات خصوصی (_).
 - پیاده سازی متدهای Getter و Setter.
 - آشنایی با متدهای ویژه مانند `__str__`.
- **هفته ۶: وراثت (Inheritance)**
 - ایجاد سلسله مراتب بین کلاس ها (کلاس والد و فرزند).
 - استفاده از `super()` و بازنویسی متدها (Method Overriding).
- **هفته ۷: چندریختی (Polymorphism) و انتزاع (Abstraction)**
 - درک مفهوم چندریختی در عمل.
 - آشنایی با کلاس ها و متدهای انتزاعی (Abstract Classes).
- **هفته ۸: کار با فایل ها در بستر شیء گرای**
 - ذخیره و بازیابی دائمی اطلاعات اشیاء در فایل های متنی (مانند CSV یا JSON).

بخش دوم: مباحث پیشرفته و پروژه عملی (هفته ۹ تا ۱۶)

- **هفته ۹: آزمون میان ترم**
 - ارزیابی عملی مفاهیم تدریس شده در هشت هفته اول.
- **هفته ۱۰: مدیریت خطا (Exception Handling)**
 - استفاده از بلوک های `try...except` برای ساخت برنامه های مقاوم در برابر خطا.
 - ایجاد خطاهای سفارشی با `raise`.
- **هفته ۱۱: مقدمه ای بر رابط کاربری گرافیکی (GUI)**
 - آشنایی با کتابخانه Tkinter.
 - ایجاد اولین پنجره و کار با ویجت های پایه (دکمه، برچسب، کادر ورودی).
- **هفته ۱۲: ساختاردهی برنامه های گرافیکی**
 - طراحی رابط کاربری به صورت شیء گرا (کلاس محور).
 - استفاده از ویجت های پیشرفته تر و مدیر چیدمان `grid`.

- **هفته ۱۳ : تکنیک‌های پیشرفته GUI**
 - ایجاد پنجره‌های چندگانه (Dialogs) و تبادل داده بین آنها.
 - **هفته ۱۴ : شروع پروژه پایانی**
 - تشریح کامل پروژه پایانی و شروع فرآیند طراحی و پیاده‌سازی تحت نظارت استاد.
 - **هفته ۱۵ : تکمیل و رفع اشکال پروژه پایانی**
 - آخرین جلسه کارگاهی برای تکمیل پروژه و رفع اشکال‌های نهایی.
 - **هفته ۱۶ : آزمون پایانی و تحویل پروژه**
 - ارزیابی نهایی و تحویل پروژه کامل شده.
-
۳. نحوه ارزیابی کارگاه:
- **تمرین‌های هفتگی (۳۰٪):** پس از هر جلسه، یک تمرین کوچک برای تحویل در جلسه بعد تعریف می‌شود.
 - **میانترم (۲۰٪):** آزمون میانترم در هفته ی نهم
 - **پروژه پایانی (۳۰٪):** جزئیات پروژه (که در جلسات آینده مشخص می‌شود) و اهمیت تحویل گام‌به‌گام آن را توضیح دهید.
 - **آزمون پایاترم (۲۰٪):** آزمون پایاترم در هفته ی نهم (شرکت در آزمون اجباری است)
 - **حضور موثر در کلاس (۲۰٪):** نمره ی اضافی
 - **نحوه تحویل:** تمرین‌ها و پروژه‌ها از طریق (ایمیل و ترجیحا با لینک GitHub) باید تحویل داده شوند.

بخش دوم: چرا پایتون؟

- خوانایی و سادگی: سیتکس نزدیک به زبان انگلیسی که یادگیری را سریع تر می کند.
- قدرت شیءگرایی: پایتون یک زبان کاملاً شیءگراست و تمام مفاهیم OOP را به خوبی پشتیبانی می کند.
- کاربردهای گسترده: اشاره به کاربرد پایتون در حوزه هایی مانند توسعه وب (Django, Flask)، هوش مصنوعی و علم داده (TensorFlow, Pandas) و اتوماسیون.
- جامعه بزرگ و پشتیبانی قوی: وجود کتابخانه های بی شمار و منابع آموزشی فراوان.

بخش سوم: نصب و راه اندازی محیط توسعه

۱. نصب مفسر پایتون (Python Interpreter):
 - آخرین نسخه پایدار پایتون را دانلود کنند.
 - نکته بسیار مهم: در هنگام نصب روی ویندوز، حتماً تیک گزینه "Add Python to PATH" را بزنید. این کار از بسیاری از مشکلات در آینده جلوگیری می کند.
 - پس از نصب، Command Prompt یا Terminal را باز کرده و با دستور `python --version` یا `python -V` از نصب صحیح آن مطمئن شوید.
۲. معرفی و نصب ویرایشگر کد (IDE):
 - استفاده از یک IDE خوب (مانند VS Code یا PyCharm) به جای Notepad ضروری است (ویژگی هایی مثل Syntax Highlighting, IntelliSense, Debugging).
 - VS Code را به عنوان یک گزینه رایگان، سبک و قدرتمند پیشنهاد میشود. و می توانید از وبسایت code.visualstudio.com آن را دانلود و نصب کنید.
۳. پیکربندی VS Code برای پایتون:
 - پس از نصب، VS Code را باز کنید.
 - به بخش Extensions بروید.
 - افزونه رسمی Python (ارائه شده توسط مایکروسافت) را جستجو و نصب کنید. این افزونه قابلیت های هوشمند را برای کدنویسی پایتون فراهم می کند.

بخش چهارم: نوشتن و اجرای اولین برنامه

۱. ایجاد فایل پروژه:

- یک پوشه جدید روی دسکتاپ خود به نام Advanced_Programming_Lab بسازید.
- در VS Code از منوی Open Folder <File این پوشه را باز کنید.
- یک فایل جدید با نام session1.py ایجاد کنید. (پسوند .py ضروری است).

۲. مرور مفاهیم پایه در عمل:

- دستور `print()`: اولین برنامه را با نمایش یک پیام ساده شروع کنید.

```
print("Hello, World!")  
print("Welcome to the Advanced Programming Lab.")
```

- متغیرها (Variables): نحوه تعریف متغیر و تخصیص مقادیر مختلف (عدد، رشته، بولین)

```
student_name = "Ali"  
student_id = 401123456  
is_active = True  
print("Student Name:", student_name)
```

- دستور `input()`: نحوه دریافت ورودی از کاربر و تعامل با برنامه

```
user_name = input("Please enter your name: ")  
print(f"Hello, {user_name}! It's nice to meet you.")
```

نکته: استفاده از f-string (مانند `f"Hello, {user_name}"`) به عنوان روشی مدرن و خوانا برای فرمت‌بندی رشته‌ها استفاده می‌شود.

- اجرای کد: نحوه اجرای فایل پایتون در VS Code (معمولاً با کلیک روی دکمه Run سبز رنگ در بالا سمت راست یا از طریق ترمینال یکپارچه)

تکلیف جلسه اول:

عنوان: برنامه محاسبه سن تقریبی.

صورت مسئله: برنامه‌ای بنویسید که نام و سال تولد کاربر را از ورودی دریافت کند. سپس سن تقریبی او را محاسبه کرده و یک پیام خوشامدگویی به همراه سن او نمایش دهد.

مثال خروجی:

```
Please enter your name: Reza
Please enter your year of birth: 1382
Hello, Reza! You are approximately 22 years old.
```

نکته راهنمایی: خروجی تابع `input()` همیشه یک رشته (`string`) است و برای محاسبات ریاضی باید آن را با استفاده از `int()` به عدد صحیح تبدیل کنید.

موفق باشید! این جلسه پایه و اساس موفقیت جلسات آینده خواهد بود.