



دانشکده مهندسی
کامپیوتر

بسمه تعالی

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

درس مبانی اینترنت اشیاء

نیمسال دوم سال تحصیلی ۱۴۰۰-۱۳۹۹

تمرین شبیه سازی سری چهارم

انجام این تمرین به صورت انفرادی است



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

مقدمه تمرین

همانطور که می دانید اینترنت اشیاء نقش به سزایی را در آسان سازی زندگی مردم، کاهش هزینه، صرفه جویی در مصرف انرژی و زمان ایفا می کند. فلذا آشنایی با پروتکل های موجود و مورد استفاده در این تکنولوژی برای درک هرچه بیشتر آن الزامیست. حال هدف از انجام این تمرین، آشنایی با پروتکل های MQTT و CoAP است، همانطور که می دانید این دو پروتکل در کاربردهای اینترنت اشیاء بسیار محبوب هستند. از دلایل استفاده از این دو پروتکل به جای سایر پروتکل های مرسوم مانند HTTP می توان به کوچک تر بودن هدر و بسته های پیام های ارسالی آن ها اشاره کرد، بدیهیست که برای کاربردهای اینترنت اشیاء ما به بسته های ارسالی کوچک تری احتیاج داریم فلذا این دو پروتکل جایگزین پروتکل مرسوم HTTP شدند. در دو لینک زیر می توانید خلاصه ای از مقایسه ای این دو پروتکل با پروتکل HTTP را برای کاربردهای اینترنت اشیاء مشاهده کنید.

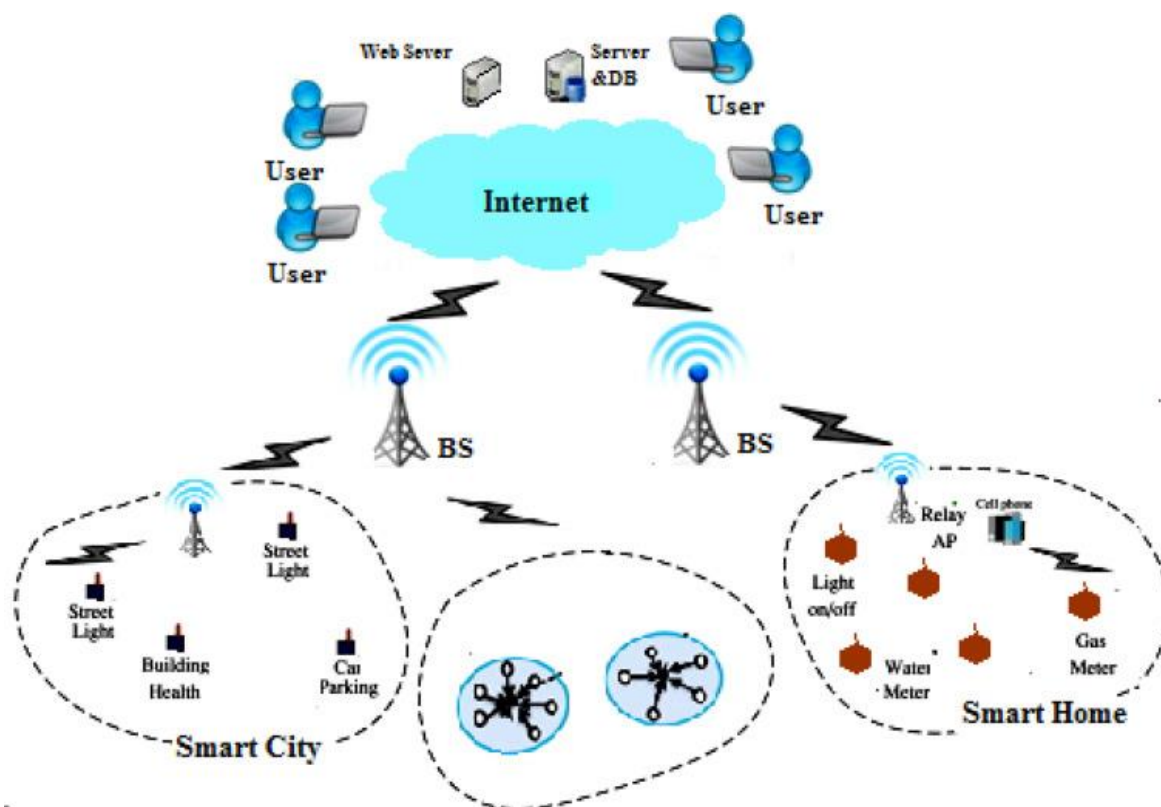
در این تمرین با کمک پروتکل های ذکر شده به شبیه سازی کاربردهای اینترنت اشیاء در زندگی واقعی مانند شهر هوشمند، کشاورزی هوشمند و ... خواهیم پرداخت. لازم به ذکر است که انتخاب این کاربردها اختیاریست و شما می توانید ۳ کاربرد دلخواه را با ایده پردازی های خود انتخاب و پیاده سازی کنید.

[مقایسه ی پروتکل های MQTT و HTTP برای کاربردهای اینترنت اشیاء](#)

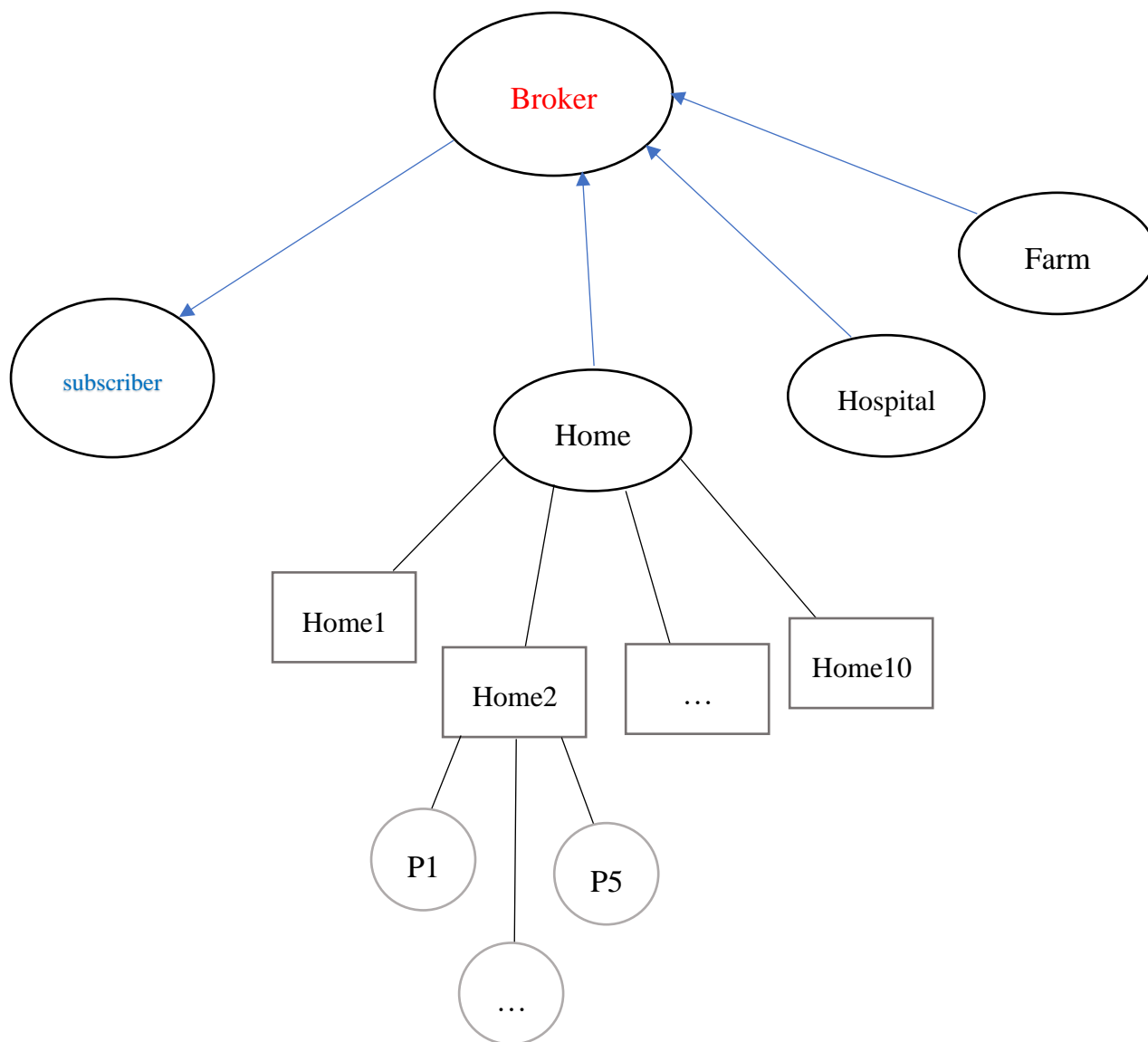
[مقایسه ی پروتکل های CoAP و HTTP برای کاربردهای اینترنت اشیاء](#)

شرح تمرین

همانطور که گفته شد اینترنت اشیا تکنولوژی نوپایی است که در جهت هوشمند سازی هرچه بیشتر زمینه‌های مختلف موجود در دنیا به کار می‌رود. این زمینه‌ها شامل شهر هوشمند، کشاورزی هوشمند، حمل و نقل هوشمند، موارد سلامتی و پزشکی هوشمند، پوشاک هوشمند و ... می‌شود. در این تمرین قصد داریم تا بر این کاربردها تمرکز کرده و با استفاده از پروتکل‌های ذکر شده، این موارد را شبیه سازی نماییم. شما در پیاده سازی این تمرین باید سه کاربرد از کاربردهای موجود را به دلخواه انتخاب کنید و سپس به پیاده سازی اجزای آن‌ها بپردازید. مقصود از این امر پیاده سازی یک شبکه‌ی heterogeneous برای کاربردهای اینترنت اشیا است. برای اطلاع بیشتر از این مفهوم [اینجا](#) کلیک کنید. در شکل زیر مثالی از این نوع شبکه به تصویر کشیده شده است:



در ابتدا به توضیح کلی مراحل پیاده سازی می پردازیم و پس از آن مراحل را به صورت خلاصه ذکر خواهیم کرد. در شکل زیر معماری مورد استفاده در این تمرین را مشاهده می کنید:



همانطور که مشاهده می کنید این معماری شامل یک subscriber، یک Broker و سه Publisher است. همانطور که در قبل گفته شد این سه Publisher را شما به صورت دلخواه انتخاب می کنید. این Publisherها همان کاربردهای انتخابی شما هستند. برای مثال در اینجا خانه های هوشمند، بیمارستان های هوشمند و مزارع هوشمند انتخاب شده است. ابتدا باید بدانید که هر کدام از این Publisherها دارای حداقل ۱۰ نمونه (instance) هستند. برای مثال در شکل بالا برای Home Node، ده خانه در ساختمان داده ای دلخواه تعریف می شوند. به همین ترتیب برای هر کدام از Farm Node و Hospital Node نیز ده نمونه تعریف می گردد. هر کدام از این نمونه ها حداقل دارای ۵ پارامتر یا ویژگی (attribute) هستند. همانطور که در

شکل نشان داده شده است همه‌ی خانه‌ها دارای ۵ پارامتر (P1...P5) هستند که برای مثال در خانه‌ها یک پارامتر به نام پارکینگ هوشمند داریم، وظیفه‌ی این پارکینگ نشان دادن وجود یا عدم وجود ماشین در پارکینگ است. دیگر پارامتری که می‌توان برای خانه نام برد سیستم گرمایشی آن است که دمای خانه را بازگو می‌کند. یک پارامتر دیگر در خانه می‌تواند میزان جریان مصرفی آن خانه باشد. (باید حداقل ۵ پارامتر را در نظر بگیرید)

سپس شما برای این ۱۰ خانه داده‌هایی تقلبی می‌سازید و آن را ذخیره می‌کنید که برای ارسال به Subscriber توسط Broker مورد استفاده قرار می‌گیرند. برای مثال در اینجا ما برای پارامترهای خانه‌ها یک وضعیت تقلبی می‌سازیم:

خانه ۱ : پارکینگ : خالی ، دمای خانه: ۲۷ درجه ، جریان مصرفی : ۵۰ وات

خانه ۲ : پارکینگ : حالی ، دمای خانه: ۲۳ درجه ، جریان مصرفی : ۶۰ وات

.

.

.

خانه ۹ : پارکینگ : پر ، دمای خانه: ۳۰ درجه ، جریان مصرفی : ۴۰ وات

خانه ۱۰ : پارکینگ : خالی ، دمای خانه: ۲۷ درجه ، جریان مصرفی : ۷۰ وات

توجه داشته باشید همین روند برای بیمارستان و مزرعه و یا هر کاربرد دیگری که شما انتخاب می‌کنید به کار گرفته می‌شود.

حال به بررسی Subscriber می‌پردازیم، در این بخش شما باید وب پیجی را جهت نمایش اطلاعات publisherها در Subscriber بالا بیاورید. این وب پیج در ابتدا سه دکمه با اسم زمینه‌های انتخابی شما دارد که بر روی هر کدام از آن‌ها که کلیک کنید اطلاعات آن بخش به نمایش گذاشته می‌شود. برای مثال اگر بر روی دکمه‌ی Home کلیک شود ۱۰ خانه‌ی تعریف شده نمایش داده می‌شود و برای هر کدام از این ۱۰ خانه به طور جداگانه ۵ پارامتر نمایش داده می‌شود. یکی از این پارامترها را (پارکینگ هوشمند) در بالا مثال زدیم.

لازم به ذکر است که در بخش MQTT، Publisherها باید ماشین‌های مجازی مجزایی باشند و Subscriber کامپیوتر شخصی شما است. (Broker شما سرور آنلاین HiveMQ است)

در بخش CoAP، Broker نیز همانند Publisherها باید ماشین مجازی مجزایی باشد.

حال در ادامه به صورت خلاصه مراحل بازگو می‌شوند:

۱. ابتدا برای پیاده سازی این نوع شبکه شما باید [حداقل سه کاربرد دلخواه](#) خود را از زمینه‌های موجود انتخاب کنید. (برای مثال بیمارستان هوشمند و ...)

۲. سپس در هر کدام از این کاربردها، [حداقل ۱۰ نمونه \(instance\)](#) ذخیره کنید. برای مثال در کاربرد خانه‌ی هوشمند شما باید ۱۰ خانه را ذخیره کنید. سپس برای این نمونه‌ها [حداقل ۵ پارامتر \(attribute\)](#) را به [دلخواه انتخاب کرده](#) و شبیه سازی خود را بر پایه‌ی آن‌ها انجام دهید. برای مثال اگر خانه‌ی هوشمند را انتخاب کردید یکی از پارامترهای آن می‌تواند پارکینگ هوشمند باشد.

۳. حال با توجه به معماری داده شده شما باید در Node های کاربردهای انتخابی خود (خانه‌ی هوشمند و ...) برای هر پارامتر یا ویژگی، مقادیری تصادفی را بدست بیاورید و آن را در ساختمان داده‌ای ذخیره کنید. برای مثال در خانه‌ی هوشمند در بخش پارامتر پارکینگ هوشمند، ما ده خانه داریم که هر کدام یک پارکینگ هوشمند دارند و وضعیت وجود یا عدم وجود وسیله‌ی نقلیه را ارسال می‌کند. [این وضعیت‌ها را به طور تصادفی مقدار دهی کنید.](#)

۴. حال اطلاعات هر کدام از Node های Publisher را توسط Broker به Subscriber انتقال دهید.

۵. با استفاده از [یک وب پیج](#) به گونه‌ای که گفته شد اطلاعات را در Subscriber به نمایش بگذارید. توجه داشته باشید که در وب پیج شما ابتدا باید [سه دکمه با اسامی کاربردهای انتخابی موجود باشد](#) که با کلیک بر هر کدام از این دکمه‌ها اطلاعات گرفته شده‌ی تمامی پارامترهای آن زمینه (مانند تمامی ۱۰ خانه) به نمایش گذاشته می‌شود.

بخش اول : راه اندازی پروتکل MQTT

معماری ساختار MQTT شامل یک کارگزار (broker) است که با یک یا چند دستگاه ارتباط برقرار میکند. پایه ارتباطات بر اساس پیام‌ها و تاپیک‌هایی (subscribe/publish) است که به یک گره ارسال میشود. در این بین دیگر گره‌ها نیز میتوانند آن پیام را دریافت کنند. سرور مرکزی مسئول مدیریت شبکه و انتقال پیام‌ها است.

۱. برای استفاده از کارگزار MQTT شما از کارگزار آنلاین HiveMQ استفاده خواهید کرد. نحوه‌ی کار با این کارگزار ساده است و با جستجو راجع به آن در اینترنت اطلاعات کافی را به دست خواهید آورد. نحوه‌ی استفاده از این Broker را در سیستم خود شرح دهید.

۲. ابتدا کاربردهای انتخابی خود را گزارش داده و پارامترهای انتخابی خود را توضیح دهید.

۳. کتابخانه‌های مورد نیاز برای MQTT Client را نصب کنید و سناریو گفته شده را اجرا کنید. این پروتکل تقریباً برای اکثر زبان‌های برنامه نویسی نسخه کلاینت دارد. پیشنهاد می‌شود از زبان python استفاده کنید. گزارشی راجع به نحوه‌ی کار سناریوی خود تهیه کنید و کدهای خود را اجرا کرده و مورد بررسی قرار دهید. همانطور که ذکر شد Publisherها باید ماشین‌های مجازی مجزایی باشند و Subscriber نیز کامپیوتر شخصی شماست.

۴. پارامترهایی که در بسته ارسالی از کلاینت به سرور و برعکس وجود دارد را توضیح دهید.

بخش دوم : راه اندازی پروتکل CoAP

همانطور که اطلاع دارید پروتکل CoAP بر مبنای درخواست/پاسخ (response/request) کار می‌کند و از متدهای Delete, Put, Post, Get برای دریافت اطلاعات استفاده میکند. در این بخش شما باید سناریوی تعریف شده را با استفاده از این پروتکل پیاده‌سازی کنید.

۱. بر خلاف بخش قبلی این بار کارگزار مورد استفاده باید توسط خود شما نوشته شود. این کارگزار را بر اساس پروتکل CoAP بنویسید. نحوه‌ی کار Broker خود را شرح دهید.

۲. کتابخانه‌های مورد نیاز برای CoAP client را نصب کنید و سناریو گفته شده را اجرا کنید. این پروتکل تقریباً برای اکثر زبان‌های برنامه نویسی نسخه کلاینت دارد. پیشنهاد می‌شود از زبان python استفاده کنید. گزارشی راجع به نحوه‌ی کار سناریوی خود تهیه کنید و کدهای خود را ران کرده و مورد بررسی قرار دهید. همانطور که ذکر شد Publisherها و Broker باید ماشین‌های مجازی مجزایی باشند و Subscriber کامپیوتر شخصی شماست.

۳. پروتکل CoAP و MQTT را از لحاظ معماری، مصرف انرژی، امنیت، کیفیت سرویس، ساینز بسته ارسالی، با یکدیگر مقایسه کنید.

بخش سوم : راه اندازی پروتکل HTTP (انجام این بخش نمره ی امتیازی دارد)

پروتکل HTTP از دست تکانی (Handshaking) به منظور ارسال و دریافت اطلاعات استفاده میکند. در این روش برای شروع و پایان عملیات تبادل اطلاعات بین سرور و کلاینت، چندین درخواست و پاسخ رد و بدل میشود. لازم به ذکر است که در این فرآیند از پروتکل های IP/TCP به منظور افزایش امنیت و تضمین ارسال داده ها استفاده میشود.

۱. بر روی سیستم شخصی HTTP server را نصب کنید و IP و Port مورد استفاده در HTTP server را نشان دهید. برای HTTP server پیشنهاد میشود از Nginx استفاده شود. (لازم به ذکر است که این سرور را خودتان نیز می توانید پیاده سازی کنید).

۲. کتابخانه های مورد نیاز برای HTTP client را نصب کنید و سناریو گفته شده را اجرا کنید. پیشنهاد می شود از زبان python استفاده کنید. گزارشی راجع به نحوه ی کار سناریوی خود تهیه کنید و کدهای خود را اجرا کرده و مورد بررسی قرار دهید. همانطور که ذکر شد Publisher ها باید ماشین های مجازی مجزایی باشند و Subscriber نیز کامپیوتر شخصی شماست.

۳. دلایل جایگزین شدن پروتکل MQTT و CoAP به جای HTTP در بسیاری از کاربردهای اینترنت اشیا را توضیح دهید.

بخش سوم : راه اندازی پروتکل AMQP (انجام این بخش نمره ی امتیازی دارد)

پروتکل AMQP یا Advanced Message Queuing Protocol یک استاندارد متن باز برای انتقال پیام های بین گره ها است. معماری ساختار AMQP شامل یک broker است که با یک یا چند دستگاه ارتباط برقرار می کند. دو عملیات exchanges و queues برای انتقال اطلاعات از گره منتشر کننده ی پیام (Publish) به دریافت کننده ی پیام (Subscriber) در این پروتکل انجام می شود.

۱. بر روی سیستم شخصی AMQP broker نصب کنید. و IP و Port مورد استفاده در AMQP broker را نشان دهید. در این خصوص می توانید از هر AMQP broker متن باز استفاده کنید.

۲. کتابخانه های مورد نیاز برای AMQP client را نصب کنید و سناریو گفته شده را اجرا کنید. گزارشی راجع به نحوه ی کار سناریوی خود تهیه کنید و کدهای خود را اجرا کرده و مورد بررسی قرار دهید. همانطور که ذکر شد Publisher ها باید ماشین های مجازی مجزایی باشند و Subscriber نیز کامپیوتر شخصی شماست.

نحوه تحویل تمرین

۱. تحویل تمرین در قالب ۷ فایل ویدئویی انجام می‌شود، یعنی برای هر مرحله از ۷ مرحله توضیح داده شده در بخش قبل باید یک فایل ویدئویی جداگانه وجود داشته باشد. در هر ویدئو مشخص شود کدام مرحله از مراحل فوق در حال انجام است. توجه داشته باشید که در هر ویدئو تمامی مراحل کار و نتایج به طور کامل حداکثر در دو دقیقه شرح داده شود.
۲. در هر ویدئو باید مشخص شده باشد که این فایل متعلق به شما است. برای مثال قبل از توضیح مراحل انجام کار، یک فایل word حاوی نام فرد، شماره دانشجویی و بخش مربوطه بر روی سیستم نشان دهید که مشخص کند این ویدئو توسط شما ضبط شده است.
۳. تمرین در قالب یک فایل zip تحویل داده شود و باید برای هر مرحله، ویدئو به همراه کد وجود داشته باشد. نحوه نامگذاری فایل باید به صورت زیر باشد:
HW1 StudentNumber.zip که در آن StudentNumber شماره دانشجویی سرگروه می‌باشد. (مثال: HW1_9923110.zip)
۴. دقت کنید که حجم فایل Zip شده نهایی، حداکثر ۱۵۰ مگابایت باشد.
۵. هر مرحله را به صورت زیر نام‌گذاری نمایید. این نحوه نام‌گذاری متناسب با تمرین خواسته شده در هر مرحله است.
 - بخش اول (01-MQTT Broker, 02-MQTT usecase, 03-MQTT scenario, 04-MQTT packets)
 - بخش دوم (01-CoAP broker, 02- CoAP scenario, 03- CoAP Comparison)
۶. بخش‌های اختیاری این تمرین به صورت آنلاین تحویل گرفته می‌شود. زمان تحویل و قواعد آن متعاقباً اعلام می‌شود.
۷. تمامی ویدئوهای ضبط شده باید قابل پخش با آخرین نسخه نرم‌افزار KMPlayer باشد.
۸. می‌توانید تمرین را به صورت گروهی انجام دهید.
۹. مهلت تحویل تمرین ۵ تیر ۱۴۰۰ است.
۱۰. به ازای هر روز تأخیر ۵ درصد جریمه در نظر گرفته خواهد شد.
۱۱. در صورت عدم رعایت موارد ذکر شده، نمره مربوط به بخش خوانایی کسر خواهد شد.