

# SER 321 B Session

**SI Session**

**Sunday, April 6th 2025**

*7:00 pm - 8:00 pm MST*

# Agenda



3-2 Protocol Tips

Communication Models

Sockets!

Review Socket Steps

Port Examination

# SI Session Expectations

Thanks for coming to the **SER 321** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
  - [tutoring.asu.edu](https://tutoring.asu.edu)
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

# Interact with us:

## Zoom Features



### Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

3-2 → Custom Protocol!

Stay Organized!

Try to emulate the structure in 3-1

Format your Markdown!

***Check out the  
recording for  
the  
discussion!***

# SER 321

## Protocol Tips

## Protocol Structure

```
## Protocol: ##
```

```
### Echo: ###
```

Request:

```
{
  "type" : "echo", -- type of request
  "data" : <String> -- String to be echoed
}
```

Error response:

```
{
  "type" : "echo",
  "ok" : false,
  "message" : <String> -- what went wrong
}
```

General response:

```
{
  "type" : "echo", -- echoes the initial response
  "ok" : <bool>, -- true or false depending on request
  "echo" : <String>, -- echoed String if ok true
  "message" : <String>, -- error message if ok false
}
```

Success response:

```
{
  "type" : "echo",
  "ok" : true,
  "echo" : <String> -- the echoed string
}
```

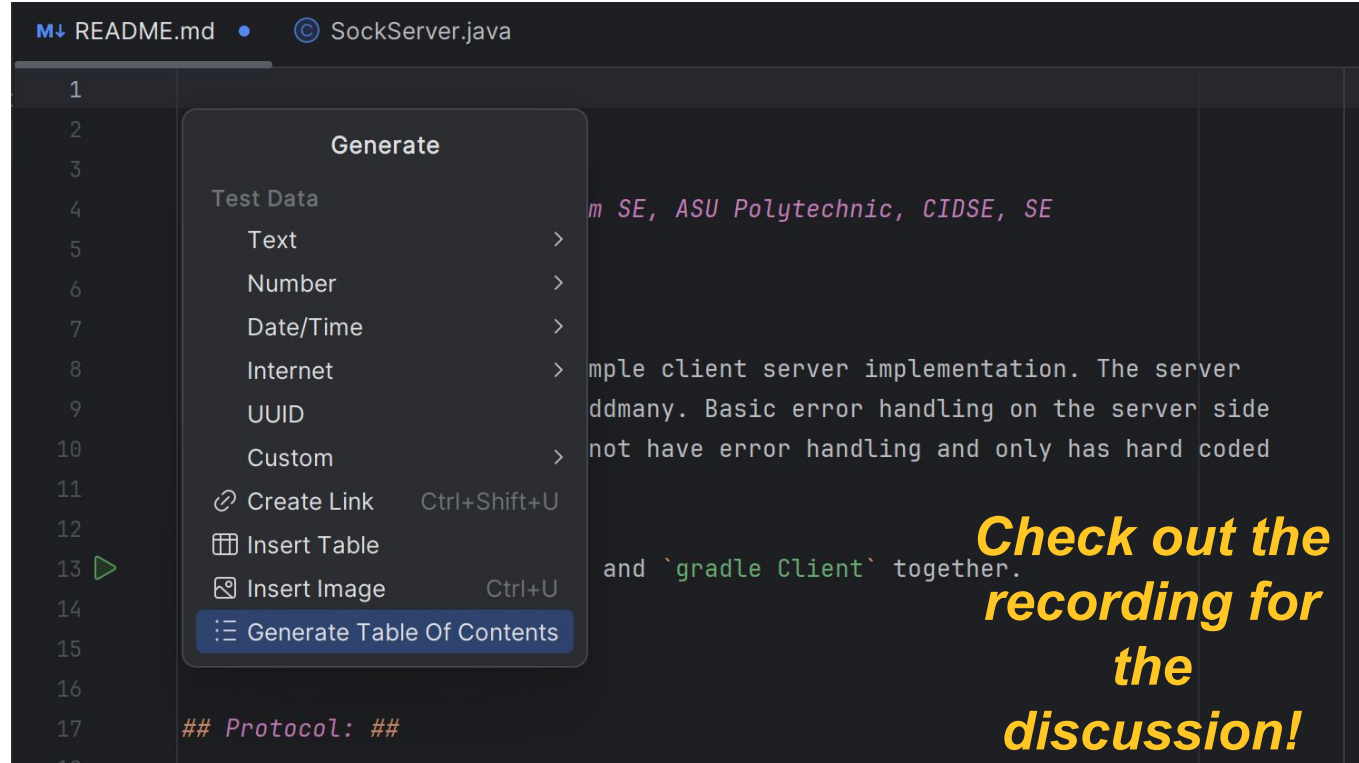
**Check out the  
recording for  
the  
discussion!**

# SER 321

## Protocol Tips

## Table of Contents

```
<!-- TOC -->
* [Protocol:](#protocol-)
* [Echo:](#echo-)
* [Add:](#add-)
* [AddMany:](#addmany-)
* [Roller:](#roller-)
* [Inventory:](#inventory-)
* [General error responses:]
<!-- TOC -->
```



The screenshot shows an IDE with a file named `README.md` and `SockServer.java`. A context menu is open over the code, titled "Generate". The menu items are:

- Test Data
  - Text >
  - Number >
  - Date/Time >
  - Internet >
  - UUID >
  - Custom >
- Create Link (Ctrl+Shift+U)
- Insert Table
- Insert Image (Ctrl+U)
- Generate Table Of Contents

The background code in `SockServer.java` includes comments like "m SE, ASU Polytechnic, CIDSE, SE" and "mple client server implementation. The server", and code snippets like "ddmany. Basic error handling on the server side" and "not have error handling and only has hard coded".

**Check out the  
recording for  
the  
discussion!**

# SER 321

## Protocol Tips

## Formatting

```
<!-- TOC -->
* [Protocol:](#protocol-)
* [Echo:](#echo-)
* [Add:](#add-)
* [AddMany:](#addmany-)
* [Roller:](#roller-)
* [Inventory:](#inventory-)
* [General error responses:]
<!-- TOC -->
```

```
## Protocol: ##
```

```
### Echo: ###
```

Request:

```
{
  "type" : "echo", -- type of request
  "data" : <String> -- String to be echoed
}
```

```
####General response:####
```

*Check out the  
recording for  
the  
discussion!*

Spaces are important!

## Protocol:

### Echo:

Request:

```
{
  "type" : "echo", -- ty
  "data" : <String> --
}
```

```
####General response:####
```



# SER 321

## Protocol Tips

## Formatting

```
<!-- TOC -->
* [Protocol:](#protocol-)
* [Echo:](#echo-)
  * [Request:](#request-)
  * [General response:](#gen
* [Add:](#add-)
* [AddMany:](#addmany-)
* [Roller:](#roller-)
* [Inventory:](#inventory-)
* [General error responses:]
<!-- TOC -->
```

```
## Protocol: ##
```

```
### Echo: ###
```

```
#### Request: ####
```

```
{
  "type" : "echo", -- type of request
  "data" : <String> -- String to be echoed
}
```

```
#### General response: ####
```

**Check out the  
recording for  
the  
discussion!**

## Protocol:

### Echo:

#### Request:

```
{
  "type" : "echo",
  "data" : <String>
}
```

#### General response:

Number of #s indicate the header level

# SER 321

## JSON Structure

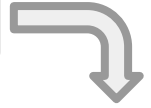
Data is stored in...

Name:Value pairs

AKA

Members

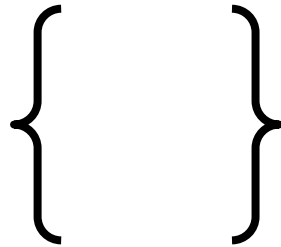
"Katie"



"student" : "Katie"

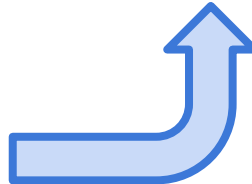
What uses curly braces?

Objects



What do Objects contain?

Members



***Check out the  
recording for  
the  
discussion!***

# SER 321

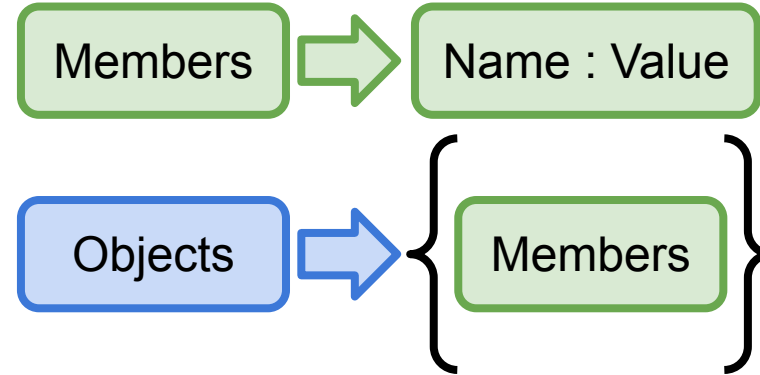
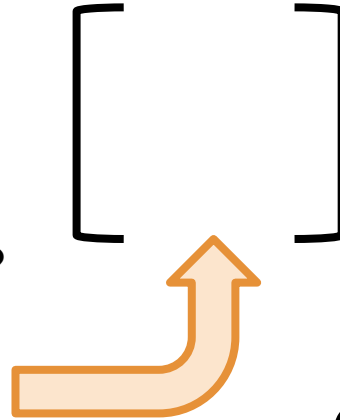
## JSON Structure

What uses brackets?

Arrays

What do Arrays contain?

Any ***Valid*** Value



***Check out the  
recording for  
the  
discussion!***



# SER 321

## JSON Structure

What is a valid value?

Strings

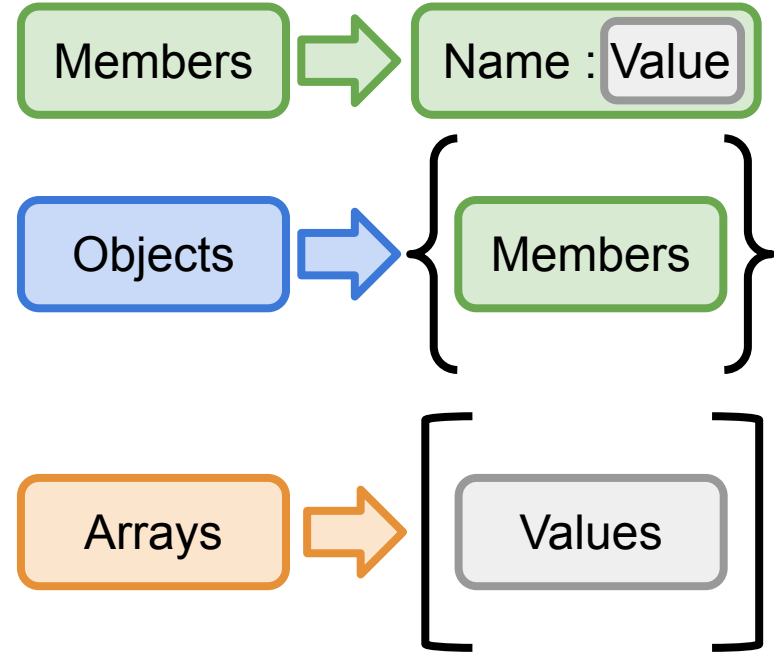
Booleans

Numbers

NULL

Objects

Arrays



***Check out the  
recording for  
the  
discussion!***

# SER 321

## Socket Protocol Types

### Two Main Conversation Models

1. Client Request

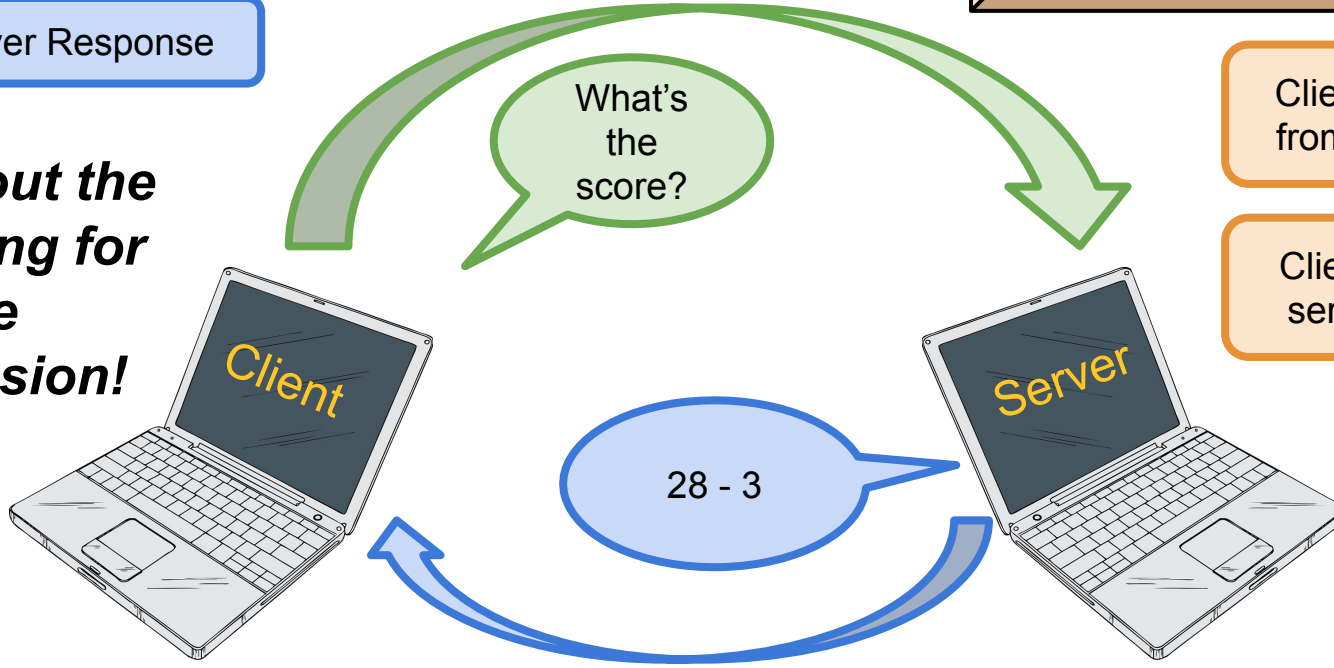
2. Server Response

#### Pull/Polling Model

Client *pulls* info from the server

Client *polls* the server for info

**Check out the recording for the discussion!**



# SER 321

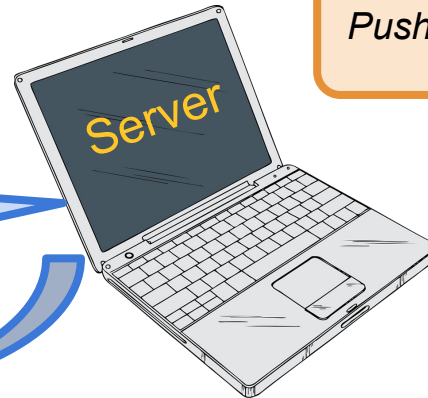
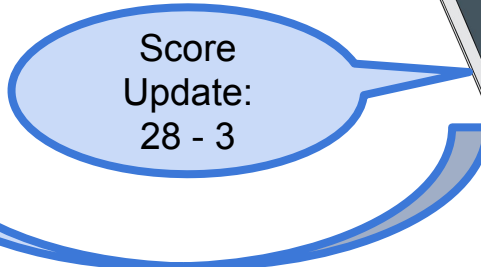
## Socket Protocol Types

### Two Main Conversation Models

1. Server sends update

2. Client acknowledges

**Check out the  
recording for  
the  
discussion!**



### Push Model

Server *pushes*  
info to client

*Push notifications*

# SER 321

## Socket Protocol Types

### Two Main Conversation Models

#### Client Push Model

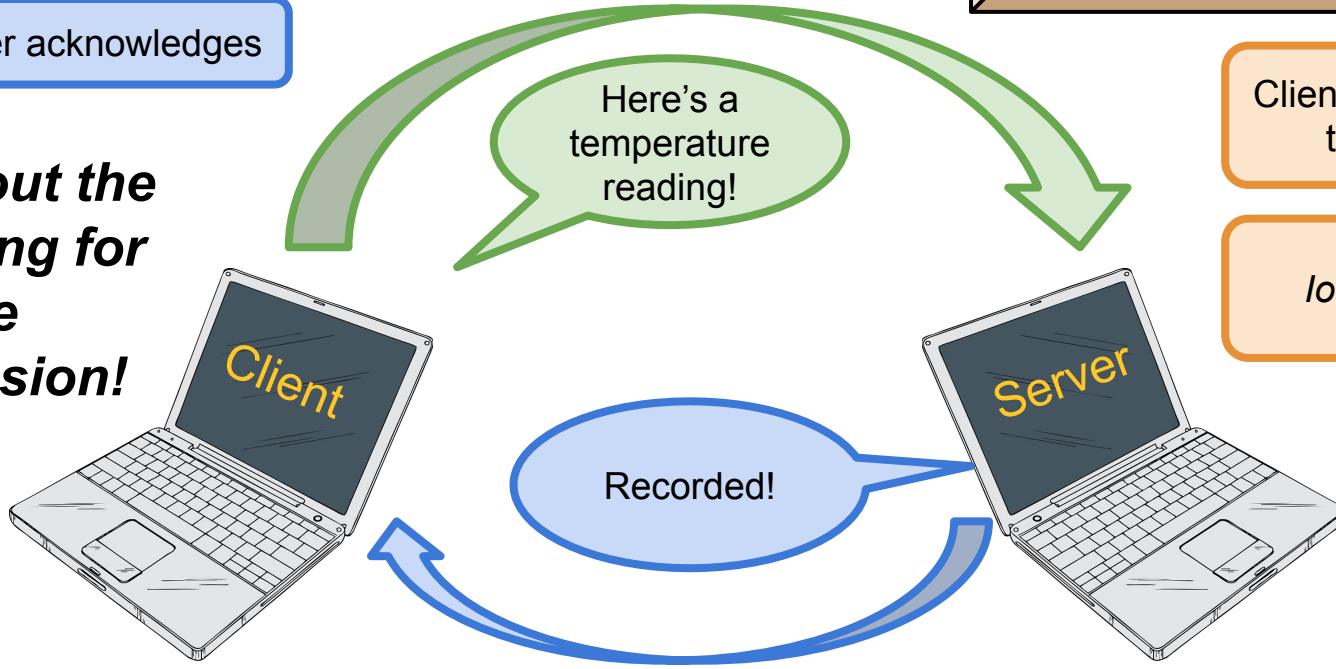
1. Client sends update

2. Server acknowledges

Client *pushes* info to Server

IoT sensors

**Check out the recording for the discussion!**



**SER 321**

**Client Socket**

Put the Steps for the **Client Socket** in the correct order:

***Check out the recording for the solution!***

1.

2.

3.

4.

5.

6.

7.

8.

- A. Send Message
- B. Close Socket
- C. Define Params
- D. Create Param Struct
- E. Receive Message
- F. Create Socket
- G. Repeat
- H. Establish Connection



## SER 321

### Server Socket

Put the Steps for the **Server Socket** in the correct order:

***Check out the recording for the solution!***

1.

2.

3.

4.

5.

6.

7.

8.

9.

- A. Mark Socket to Listen
- B. Close Socket
- C. Define Params
- D. Create Param Struct
- E. Continue Listening
- F. Handle Client
- G. Wait for Connection
- H. Bind Socket to Address
- I. Create Socket

# SER 321

Scratch Space

## Upcoming Events

### SI Sessions:

- Tuesday, April 8th at 10:00 am MST
- Thursday, April 10th at 7:00 pm MST
- Sunday, April 13th at 7:00 pm MST

### Review Sessions:

- Sunday, April 27th at **6:00 pm MST - 2 hour Exam Review Session**
- Tuesday, April 29th, at 10:00 am MST - **Q&A Session**

# Questions?

## Survey:

<https://asuasn.info/ASNSurvey>



# More Questions?

Check out our other resources!

tutoring.asu.edu



## Academic Support

Academic Support Network (ASN) provides a variety of free services in-person and online to help currently enrolled ASU students succeed academically.

### Services



#### Subject Area Tutoring

Need in-person or online help with math, science, business, or engineering courses? Just hop into our Zoom room or drop into a center for small group tutoring. We'll take it from there.

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)

Go to Zoom



#### Writing Tutoring

Need help with undergraduate or graduate writing assignments? Schedule an in-person or online appointment, access your appointment link, or wait in our drop-in queue.

[Access your appointment link](#)

[Access the drop-in queue](#)

Schedule Appointment



#### Online Study Hub

Join our online peer communities to connect with your fellow Sun Devils. Engage with our tools to search our bank of resources, videos, and previously asked questions. Or, ask our Tutorbot questions.

Now supporting courses in Math, Science, Business, Engineering, and Writing.

Online Study Hub

1-

Go to Zoom

2-

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)



1. Click on 'Go to Zoom' to log onto our Online Tutoring Center.
2. Click on 'View the tutoring schedule' to see when tutors are available for specific courses.

# More Questions?

## Check out our other resources!

[tutoring.asu.edu/online-study-hub](https://tutoring.asu.edu/online-study-hub)

 **Academic Support Network**

 [Services](#) [Faculty and Staff Resources](#) [About Us](#)

[University College](#)

## Online Study Hub

Online peer communities for students and tutors, YouTube channels, and Tutorbots.



### What are online peer communities?

Individual courses have an online peer community that allows you to connect with your peers to post and answer questions and to develop study groups.



### How can tutoring center videos help?

Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.



### How does the Tutorbot work?

You can ask the Tutorbot questions about course concepts and the Tutorbot will recommend additional resources and examples to help address your questions.

Select a subject

- Any -

Apply



Academic Support Network



[Services](#)

[Faculty and Staff Resources](#)

[About Us](#)

[University College](#)

Select a subject

- Any -

Apply

Business

### ACC 231

Uses of Accounting Info I

 [Peer Community](#)


### ACC 241

Uses of Accounting Info II

 [Peer Community](#)

### CIS 105

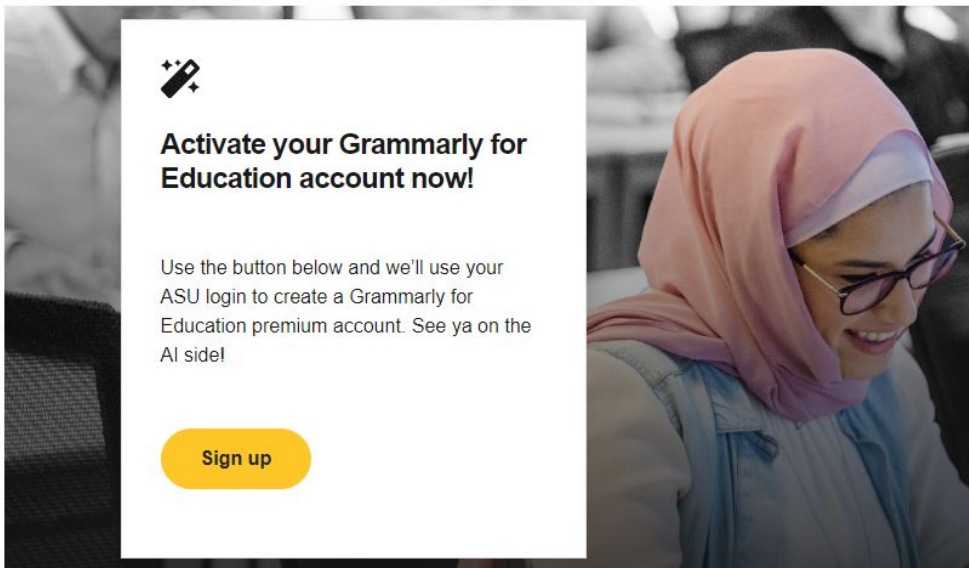
Computer Applications and Information Technology


 [Peer Community](#)

Don't forget to check out the Online Study Hub for additional resources!

# Expanded Writing Support Available

Including Grammarly for Education, at no cost!





**Activate your Grammarly for Education account now!**

Use the button below and we'll use your ASU login to create a Grammarly for Education premium account. See ya on the AI side!

[Sign up](#)



[tutoring.asu.edu/expanded-writing-support](https://tutoring.asu.edu/expanded-writing-support)

\*Available slots for this pilot are limited

## Additional Resources

- [Course Repo](#)
- [Gradle Documentation](#)
- [GitHub SSH Help](#)
- [Linux Man Pages](#)
- [OSI Interactive](#)
- [MDN HTTP Docs](#)
  - [Requests](#)
  - [Responses](#)
- [JSON Guide](#)
- [org.json Docs](#)
- [javax.swing package API](#)
- [Swing Tutorials](#)