SER 321 B Session

SI Session

Thursday, April 3rd 2025

7:00 pm - 8:00 pm MST

Agenda

- OSI Model Review
- Sockets!
  - Properties & Steps for Use
  - Handling the Client
  - 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
- 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

SER 321
Sockets!

Sockets allow our client and server to communicate!

Need to define **3 properties** before usage

- Location
- Connection Semantics
- Message Format

Check out the recording for the discussion!

IP or DNS	142.251.46.206	www.google.com
TCP or UDP	Connection Oriented	Connectionless
Protocol Specs	Synchronous	Asynchronous
	Stateless	Stateful
	Binary	Text
	Headers	No Headers

SER 321
Sockets!

Sockets allow our client and server to communicate!

Person

l to define **3 properties** before usage

Conversation Flow

Conversation Content

Check out the recording for the discussion!

IP or DNS	142.251.46.206	www.google.com
TCP or UDP	Connection Oriented	Connectionless
Protocol Specs	Synchronous	Asynchronous
	Stateless	Stateful
	Binary	Text
	Headers	No Headers

SER 321
Client Socket

Steps for the **Client Socket**

1.
2.
3.
4.
5.
6.
7.
8.

Check out the recording for the discussion and solution!

SER 321
Server Socket

Steps for the **Server Socket**

1.
2.
3.
4.
5.
6.
7.
8.
9.

Check out the recording for the discussion and solution!

Assign 3-1 Starter Code

SER 321

Server Socket

1. Define Params
2. Create Socket
3-5. Mark Socket to Listen
6. Wait for Connection
7. Handle Client Connection
8. Close Client Connection
9. Continue Listening

```
public static void main (String args[]) {

  if (args.length != 1) {
    System.out.println("Expected arguments: <port(int)>");
    System.exit( status: 1);
  }

  try {
    port = Integer.parseInt(args[0]);
  } catch (NumberFormatException nfe) {
    System.out.println("[Port|sleepDelay] must be an integer");
    System.exit( status: 2);
  }

  try {
    //open socket
    ServerSocket serv = new ServerSocket(port);
    System.out.println("Server ready for connections");

    /** Simple loop accepting one client and calling handling one request. */

    while (true){
      System.out.println("Server waiting for a connection");
      sock = serv.accept(); // blocking wait
      System.out.println("Client connected");
```

1

2 & 3-5

9

6

7

8

Check out the recording for the discussion and solution!

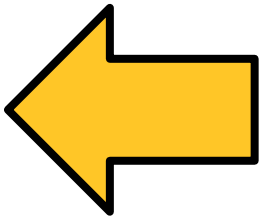
Assign 3-1 Starter Code

SER 321
Server Socket

What needs to be done here?

Check out the recording for the discussion and solution!

1. Define Params
2. Create Socket
3-5. Mark Socket to Listen
6. Wait for Connection
7. Handle Client Connection
8. Close Client Connection
9. Continue Listening



1

2

3

4

5

Assign 3-1 Starter Code

Echo_Java

SER 321

Server Socket

What needs to be done here?

Check out the recording for the discussion and solution!

Is input ***from the client*** or ***to the client*** ?

1

2

3

4

5

1. Define Params

```
// setup the object reading channel
in = new ObjectInputStream(sock.getInputStream());

// get output channel
OutputStream out = sock.getOutputStream();

// create an object output writer (Java only)
os = new DataOutputStream(out);
```

```
clientSock = sock.accept(); // blocking wait
PrintWriter out = new PrintWriter(clientSock.getOutputStream(), autoFlush: true);
InputStream input = clientSock.getInputStream();
System.out.println("Server connected to client");
```

Assign 3-1 Starter Code

SER 321

Server Socket

What needs to be done here?

Check out the recording for the discussion and solution!

```
static void overandout() {
  try {
    os.close();
    in.close();
    sock.close();
  } catch(Exception e) {e.printStackTrace();}

}
```

```
try {
  s = (String) in.readObject();
} catch (Exception e) {
  System.out.println("Client disconnect");
  connected = false;
  continue;
}
```

1. Create input/output streams
2.
3.
4.
5.

Assign 3-1 Starter Code

SER 321

Server Socket

What needs to be done here?

Check out the recording for the discussion and solution!

```
JSONObject res = isValid(s);

if (res.has( key: "ok")) {
  writeOut(res);
  continue;
}

JSONObject req = new JSONObject(s);

res = testField(req,  key: "type");
if (!res.getBoolean( key: "ok")) {
  res = noType(req);
  writeOut(res);
  continue;
}
```

```
public static JSONObject isValid(String json) {
  try {
static JSONObject testField(JSONObject req, String key){
  JSONObject res = new JSONObject();

  // field does not exist
  if (!req.has(key)){
    res.put("ok", false);
    res.put("message", "Field " + key + " does not exist in request");
    return res;
  }
  return res.put("ok", true);
}
      return res;
    }
  }
  return new JSONObject();
}
```

Assign 3-1 Starter Code

Echo_Java

SER 321

Server Socket

What needs to be done here?

Check out the recording for the discussion and solution!

```
int numr = input.read(clientInput,  off: 0, bufLen);
```

```
String received = new String(clientInput,  offset: 0, numr);
System.out.println("read from client: " + received);
out.println(received);
```

```
if (req.getString( key: "type").equals("echo")) {
  res = echo(req);
} else if (req.getString( key: "type").equals("add")) {
  res = add(req);
} else if (req.getString( key: "type").equals("addmany")) {
  res = addmany(req);
} else {
  res = wrongType(req);
}
writeOut(res);
```

1. Create input/output streams
2. Check for disconnect
3. Check Protocol
4.
5.

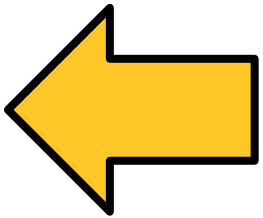
Assign 3-1 Starter Code

SER 321
Server Socket

What needs to be done here?

Check out the recording for the discussion and solution!

1. Define Params
2. Create Socket
3-5. Mark Socket to Listen
6. Wait for Connection
7. Handle Client Connection
8. Close Client Connection
9. Continue Listening



1 Create input/output streams

2 Check for disconnect

3 Check Protocol

4 Read Headers

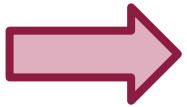
5 Handle Accordingly

Check out the recording for the discussion!

SER 321

OSI Model

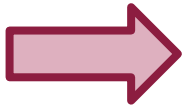
Unit	Layer	What we are *really* talking about
Data		
Data		
Data		
Segment		
Packet		
Frame		
Bits		



Check out the recording for the discussion!

SER 321

OSI Model

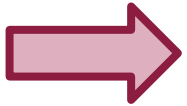


Unit	Layer	What we are *really* talking about
Data		
Data		
Data		
Segment		
Packet		
Frame		
Bits	Physical	Signal, Binary transmission

Check out the recording for the discussion!

SER 321

OSI Model

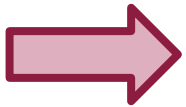


Unit	Layer	What we are *really* talking about
Data		
Data		
Data		
Segment		
Packet		
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

Check out the recording for the discussion!

SER 321

OSI Model

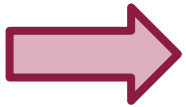


Unit	Layer	What we are *really* talking about
Data		
Data		
Data		
Segment		
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

Check out the recording for the discussion!

SER 321

OSI Model

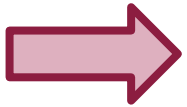


Unit	Layer	What we are *really* talking about
Data		
Data		
Data		
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

Check out the recording for the discussion!

SER 321

OSI Model

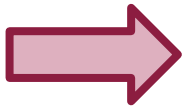


Unit	Layer	What we are *really* talking about
Data		
Data		
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

Check out the recording for the discussion!

SER 321

OSI Model

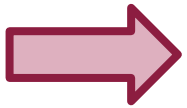


Unit	Layer	What we are *really* talking about
Data		
Data	Presentation	Translation, compression, encryption
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

Check out the recording for the discussion!

SER 321

OSI Model



Unit	Layer	What we are *really* talking about
Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	Translation, compression, encryption
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

SER 321

Scratch Space

Upcoming Events

SI Sessions:

- Sunday, April 6th at 7:00 pm MST
- Tuesday, April 8th at 10:00 am MST
- Thursday, April 10th 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:

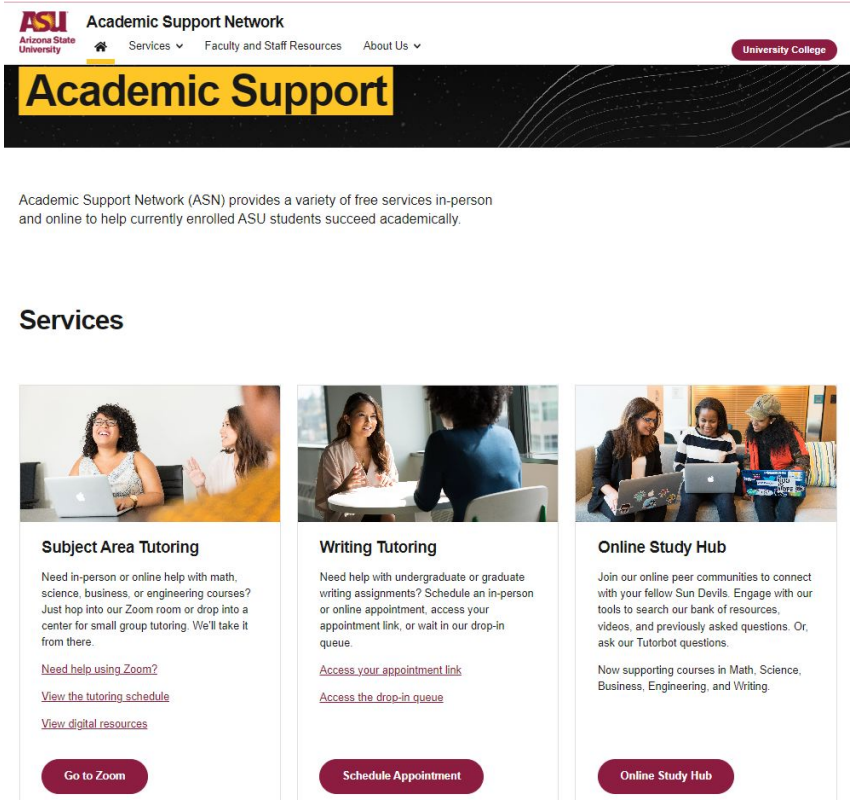
https://asuasn.info/ASNSurvey

26

More Questions?

Check out our other resources!

tutoring.asu.edu

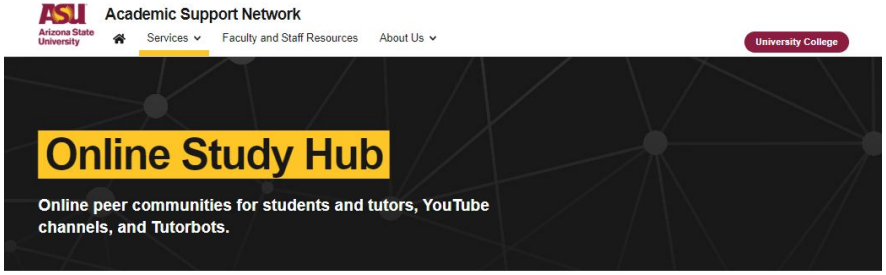


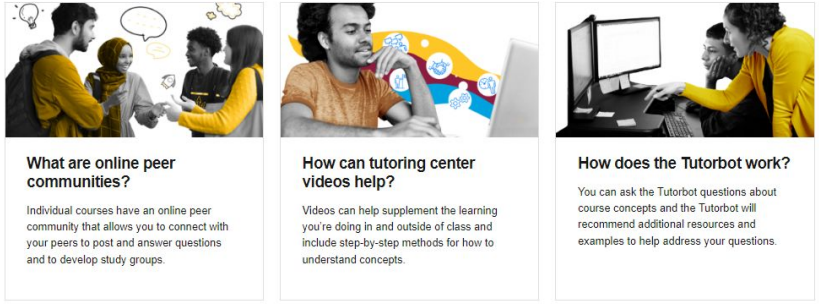
1. **Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.**
2. **Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.**

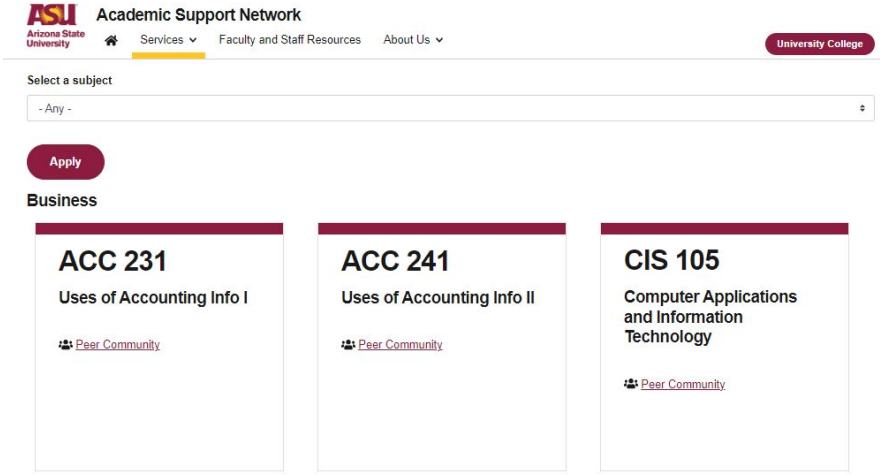
More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub



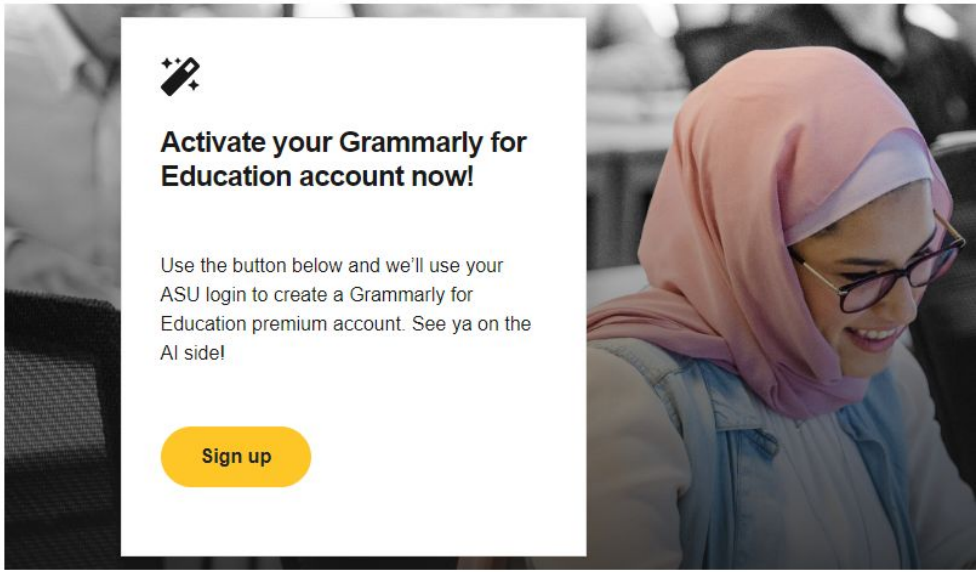




Don't forget to check out the Online Study Hub for additional resources!

Expanded Writing Support Available

Including Grammarly for Education, at no cost!



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