SER 321 A Session

SI Review Session

Monday October 2nd 2023

6:00 pm - 8:00 pm MST

Agenda

- Networking Upper and Lower Layers
- Programming with Sockets
- Threads and Serialization
- Distributed Algorithms
- Middleware

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
Preparing for the Final

Katie's Tips:

- Take advantage of the allowed reference sheet!
    - One sheet, front and back allowed
    - **Handwritten**
- Review the quizzes
    - Review the problems you struggled with
    - If you think you will forget it, add it to the sheet!
- Focus on the BIG concepts

SER 321

OSI Model

Layer	Data Unit	What we mean
Application	Data	
Presentation	Data	
Session	Data	
Transport	Segments	
Network	Packets	
Data Link	Frames	
Physical	Bits	

SER 321

OSI Model

Layer	Data Unit	What we mean
Application	Data	
Presentation	Data	
Session	Data	
Transport	Segments	
Network	Packets	
Data Link	Frames	
Physical	Bits	

SER 321

OSI Model

Layer	Data Unit	What we mean
Application	Data	
Presentation	Data	
Session	Data	
Transport	Segments	
Network	Packets	
Data Link	Frames	
Physical	Bits	

SER 321

OSI - Critical Middle Layers

Layer	Data Unit	What we mean
Application	Data	HTTP(s), SMTP, IMAP, POP, FTP
Presentation	Data	Translate, compress, encrypt
Session	Data	Authentication, authorization, session management
Transport	Segments	TCP/UDP
Network	Packets	IP Address, routing and delivery
Data Link	Frames	LLC, MAC, data transmission in LAN
Physical	Bits	Media Signal and Binary transmission

SER 321
Network Layer

We already said it's responsible for "IP address, routing and delivery", but what does that mean?

- Routing and Delivery of what?
  - Packets
- Delivery to where?
  - Target address
- Connection-Oriented or Connectionless?
  - Connectionless
  - Each packet is handled individually
- Reliable or Unreliable?
  - Unreliable
  - Going to try really hard, but *no* guarantee

SER 321
Network Layer

Network:
Subnet:
Host:
Port:

Think fast! Which of the following correctly identifies the different pieces of an IP address?

A.

128.148.32.110:8080

B.

128.148.32.110:8080

C.

128.148.32.110:8080

D.

128.148.32.110:8080

SER 321
Network Layer

Network:
Subnet:
Host:
Port:

Think fast! Which of the following correctly identifies the different pieces of an IP address?

A.

128.148.32.110:8080

B.

128.148.32.110:8080

C.

128.148.32.110:8080

D.

128.148.32.110:8080

SER 321

OSI - Critical Middle Layers

Layer	Data Unit	What we mean
Application	Data	HTTP(s), SMTP, IMAP, POP, FTP
Presentation	Data	Translate, compress, encrypt
Session	Data	Authentication, authorization, session management
Transport	Segments	TCP/UDP
Network	Packets	IP Address, routing and delivery
Data Link	Frames	LLC, MAC, data transmission in LAN
Physical	Bits	Media Signal and Binary transmission

SER 321
Transport Layer

TCP vs. UDP

Starting to discuss the actual transmission here

Both have guarantees that they provide

TCP

- Reliable data transfer
- Sequential delivery
- Three-way handshake

UDP

- No guarantees
- Distinguish data per port

TCP vs. UDP

Okay, so which is which here?

- Connectionless
- Asynchronous
- Connection-oriented
- Synchronous

TCP vs. UDP

Okay, so TCP seems pretty nice. When would you use UDP?

Speed is more important than reliability

Dropped packets are not concerning - it'll get it on the next cycle

You plan to implement your own custom QOS

SER 321
Application Layer

Protocol:
Host:
Path:
Query:

Think fast! Which of the following correctly identifies the different pieces of a URL?

A.

https://www.google.com/search?q=asu

B.

https://www.google.com/search?q=asu

C.

https://www.google.com/search?q=asu

D.

https://www.google.com/search?q=asu

SER 321
Application Layer

Protocol:
Host:
Path:
Query:

Think fast! Which of the following correctly identifies the different pieces of a URL?

A.

https://www.google.com/search?q=asu

B.

https://www.google.com/search?q=asu

C.

https://www.google.com/search?q=asu

D.

https://www.google.com/search?q=asu

SER 321
Socket Programming

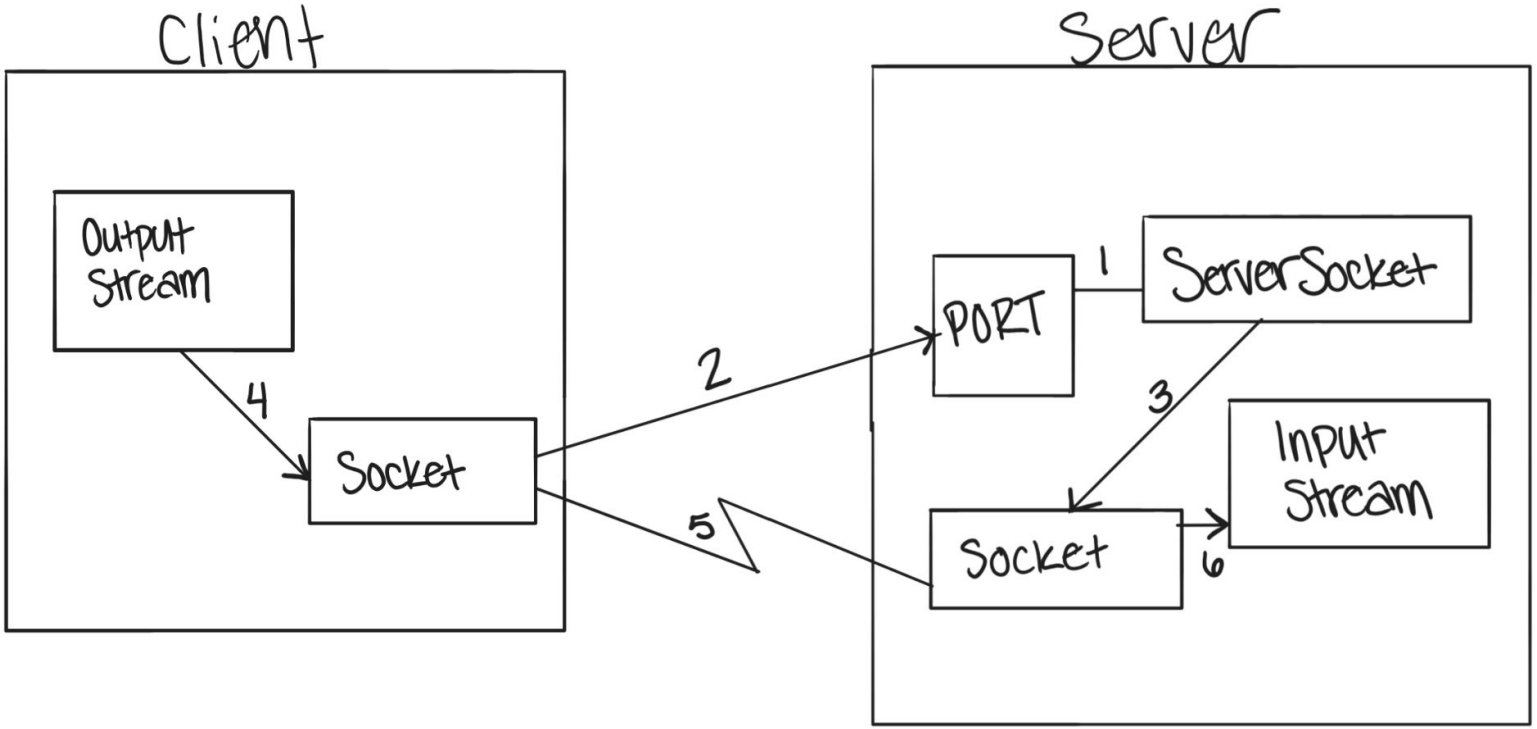
Can we name the three Socket Properties?

1. Location
2. Connection Semantics
3. Message Format

1. IP address, or use DNS to get the IP from the name
2. Connectionless or Connection-oriented
   Asynchronous or Synchronous
   Stateful or Stateless
3. Protocol, Error Handling, and data format

SER 321

Socket Programming



SER 321

Threads and Synchronization

Do we remember the threading pitfalls?

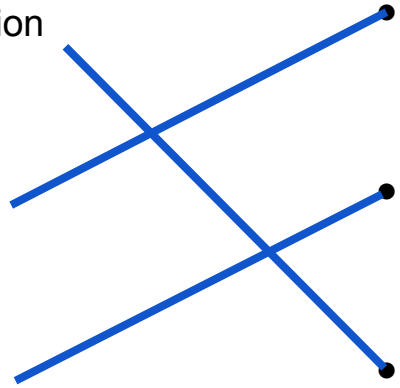
- Race Condition
  - More than one thread accesses a single resource at one time
- Starvation
  - One thread never gets access to the resource it needs
- Deadlock
  - A thread is only able to acquire access to part of its resources

SER 321

Threads and Synchronization

Do we remember the threading pitfalls?

- Race Condition
- Starvation
- Deadlock



One thread never gets access to the resource it needs

A thread is only able to acquire access to part of its resources

More than one thread accesses a single resource at one time

SER 321
Threads and Synchronization

- Locks and Semaphores
- Atomic Variables
- Volatile Keyword
- Monitor

Solutions?

What stream do I use??

- Buffered Streams
- Data Streams
- Object Streams

What stream do I use??

- Buffered Streams — Bytes
- Data Streams
- Object Streams

What stream do I use??

- Buffered Streams — Bytes
- Data Streams — Primitive Data Types
- Object Streams

What stream do I use??

● Buffered Streams	Bytes
● Data Streams	Primitive Data Types
● Object Streams	Objects

SER 321

Threads and Synchronization

JSON Object or Array?

```
{
  "name" : "labREST",
  "description" : "SER421 REST API Lab",
  "dependencies": {
    "apidoc": "^1.2.0"
  },
  "apidoc" : {
    "title" : "labREST kgrinne3",
    "url" : "http://localhost:8080",
    "header" : {
      "title" : "Introduction",
      "filename" : "README.md"
    },
    "template" : {
      "showRequiredLabels" : true
    }
  }
}
```

SER 321

Threads and Synchronization

JSON Object or Array?


```
[
  {
    "category": "Science: Computers",
    "type": "multiple",
    "difficulty": "easy",
    "question": "In web design, what does CSS stand for?",
    "correct_answer": "Cascading Style Sheet",
    "incorrect_answers": [
      "Counter Strike: Source",
      "Corrective Style Sheet",
      "Computer Style Sheet"
    ]
  },
  {
    "category": "Science: Computers",
    "type": "multiple",
    "difficulty": "easy",
    "question": "What amount of bits commonly equals one byte?",
    "correct_answer": "8",
    "incorrect_answers": [
      "1",
      "2",
      "64"
    ]
  }
]
```

SER 321
Distributed Algorithms

Can we list the issues?

- Nodes fail
- Latency
- Network communication failures
- Common and/or shared resources
- Deadlocks
- Execution safety - no bad stuff
- Liveliness - everyone goes eventually

SER 321
Distributed Algorithms

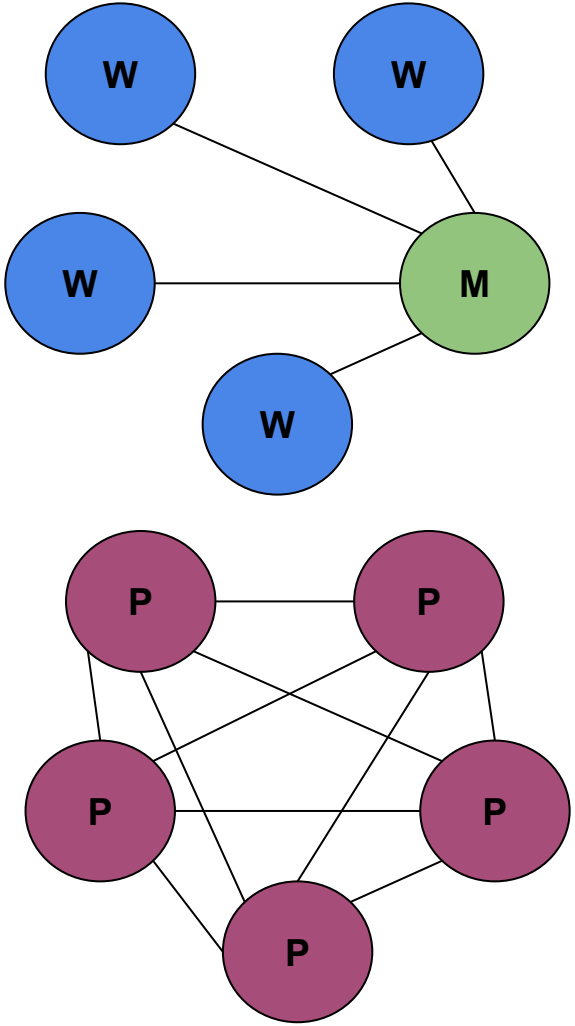
Should we always distribute the workload?

No!

Think XXXXXL!

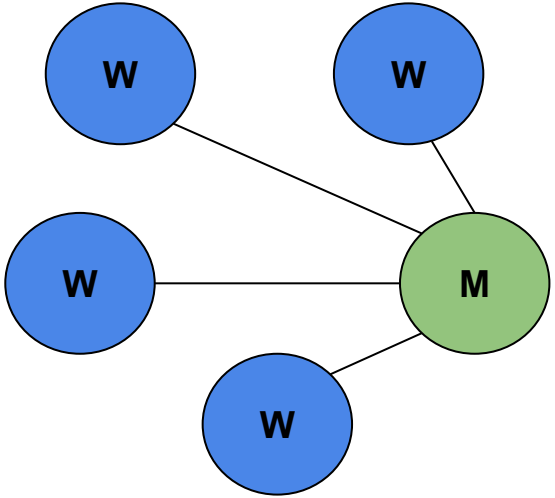
SER 321
Distributed Structures

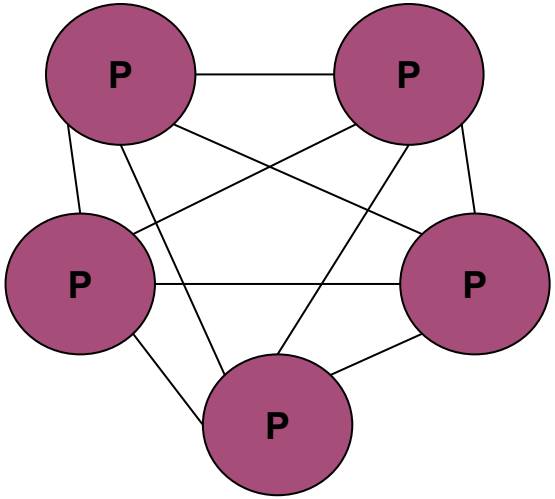
What are we looking at?



SER 321
Distributed Structures

Can we list some pros of both?

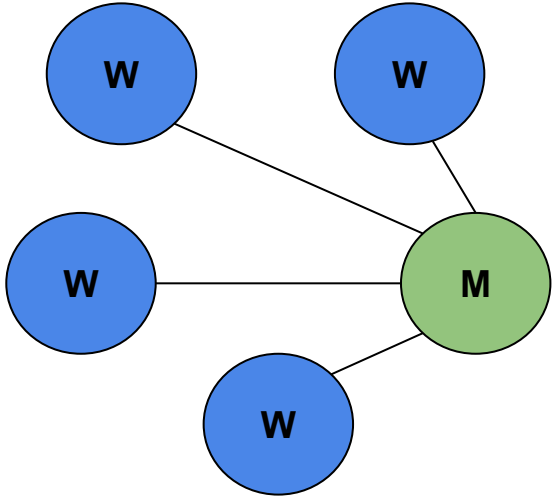




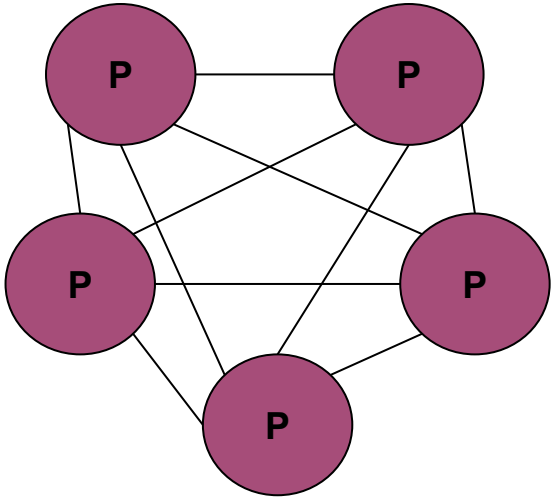
SER 321
Distributed Structures

Can we list some pros of both?

- Simple setup
- Logic is isolated
- Communication is linear

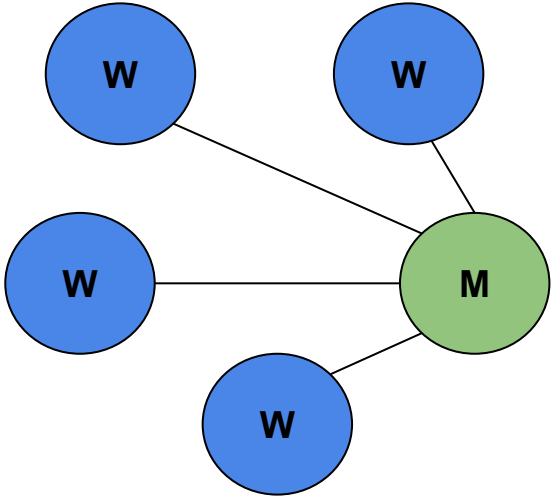


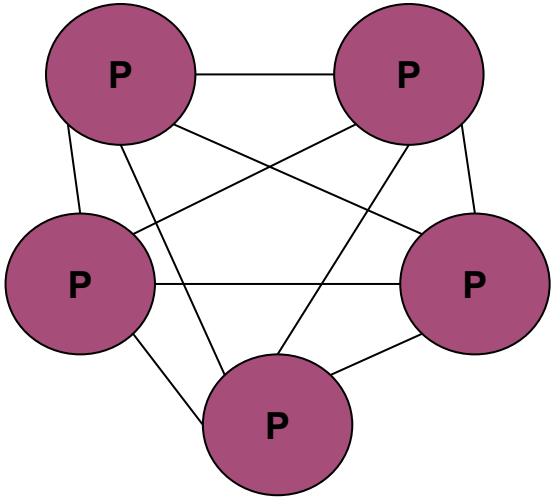
- Peers can join easily
- Peers can (and do) fail without issue
- Peers are all equal



SER 321
Distributed Structures

What about cons?

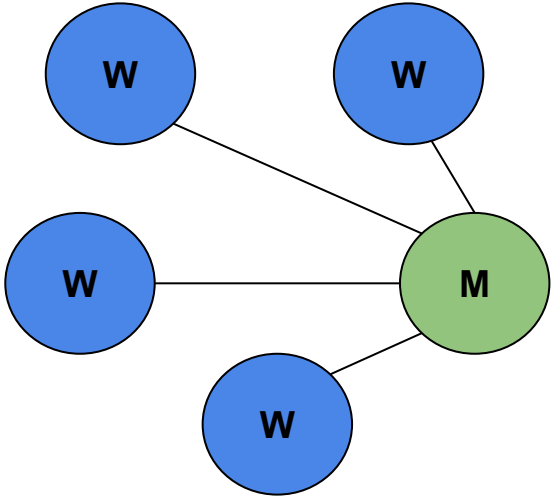




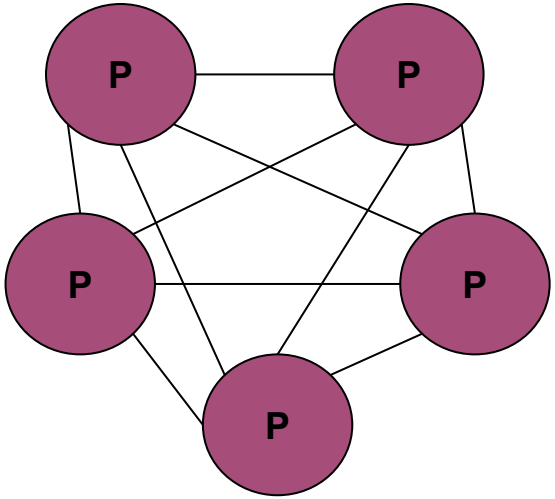
SER 321
Distributed Structures

What about cons?

- Some nodes are more important
- Single point of failure



- Client connections are different
- Communication is complicated

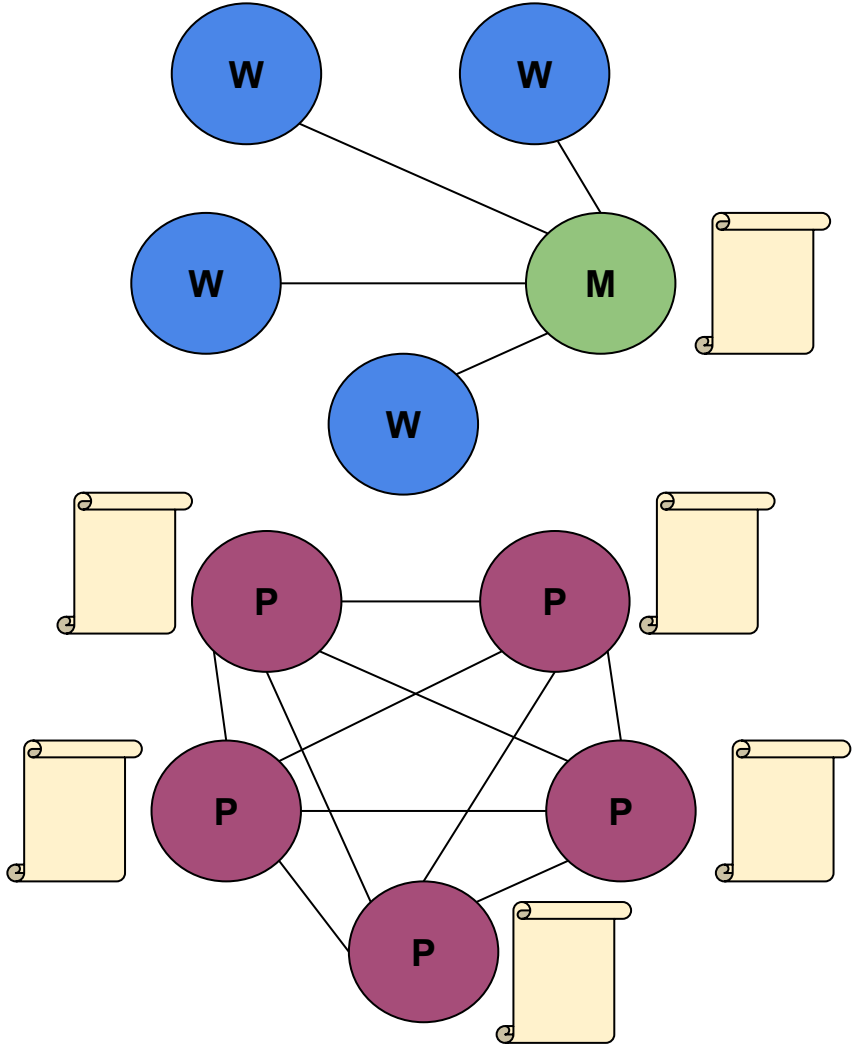


SER 321
Distributed Structures

What about cons?

- Some nodes are more important
- Single point of failure

- Client connections are different
- Communication is complicated



SER 321
Middleware

What is it?

The cool stuff!

Glue between client and server

Interface for communication

Think API

SER 321

Middleware

Benefits?

- Agility
- Efficiency
- Portability
- Reusability
- Cost Effectiveness

Questions?

Survey:

https://bit.ly/asn_survey

40

Upcoming Events

Final Exam

Opens: Wednesday October 4th

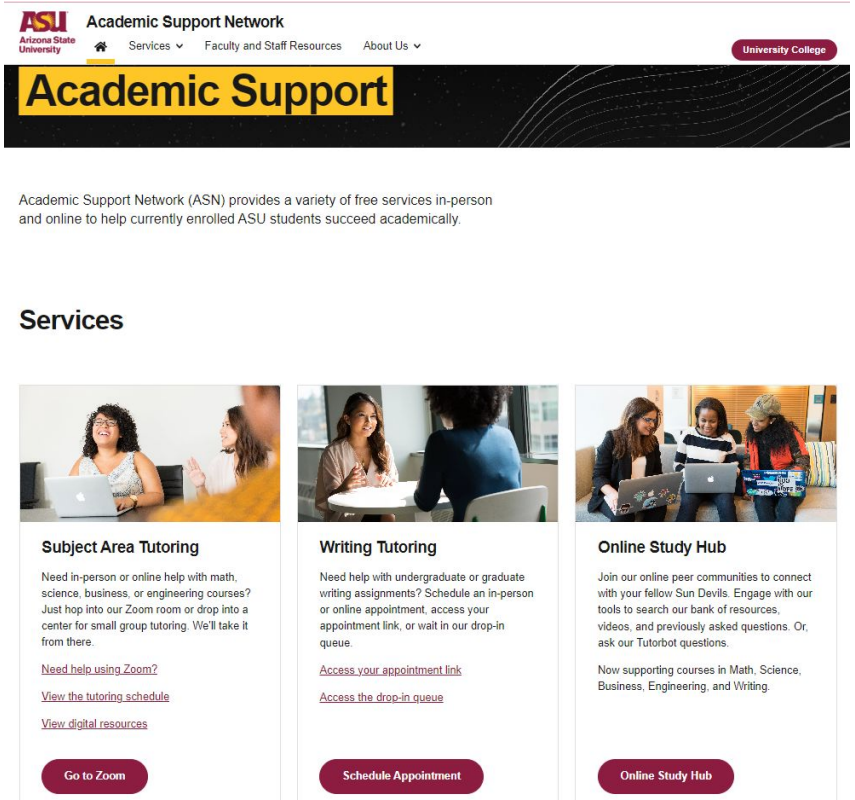
Closes: Friday October 6th at 11:59 pm MST

Good Luck!
You got this!

More Questions?

Check out our other resources!

tutoring.asu.edu

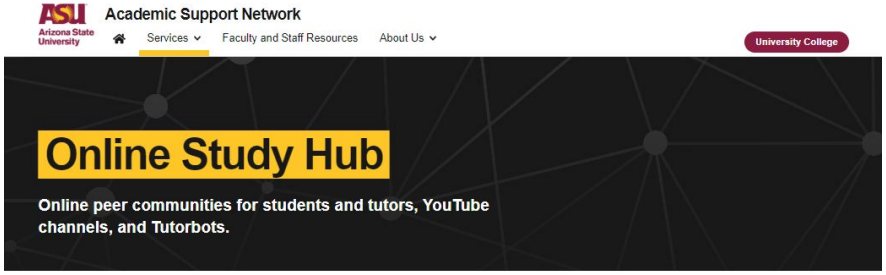


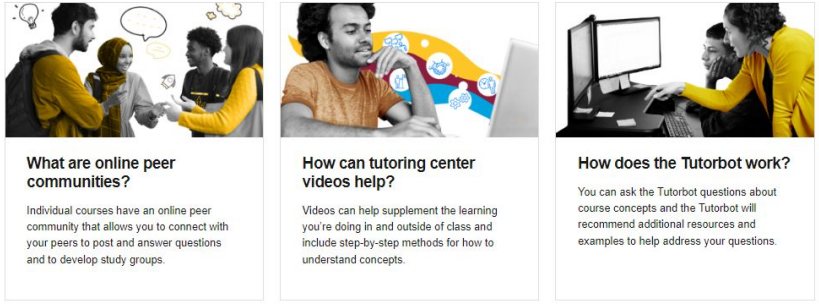
1. **Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.**
2. **Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.**

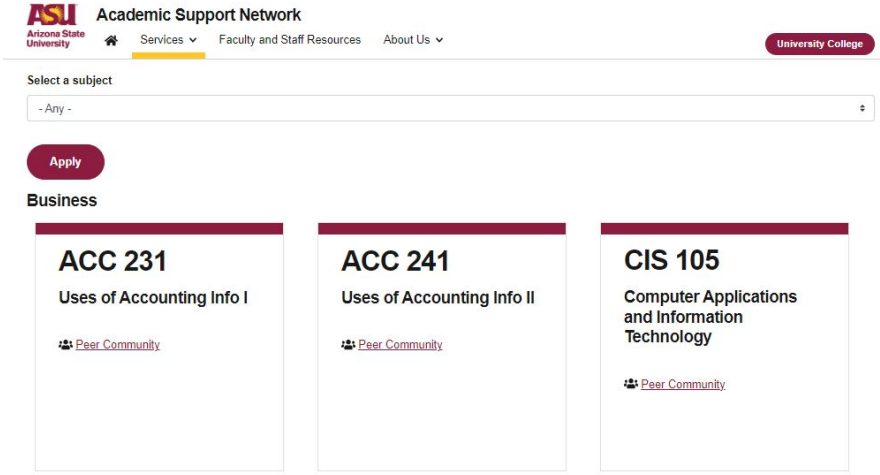
More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub







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