

# SER 321 B Session

**SI Session**

**Monday, November 6th 2023**

*4:00 - 5:00 pm MST*

# Agenda



Serialization

JSON Review

Protocol Buffers

Managing Concurrency

# 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

# SER 321

## Java IO Streams

What's the difference?

- Buffered Streams
- Data Streams
- Object Streams

*Check out the recording for the solution!*

# SER 321

## Serialization

What is it? 🤔

“Translating data structures or object states for storage or transmission”

Main Forms:

- XML
- ➡ • JSON
- Java Serialization (Objects)
- ➡ • Protocol Buffers

# SER 321

## Serialization

What is it? 🤔

“Translating data structures or object states for storage or transmission”

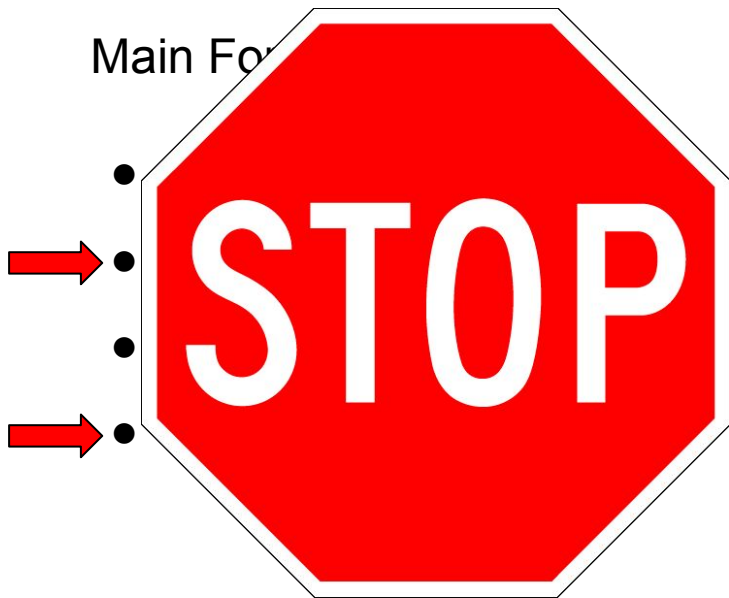
Main For

Text

or

Binary

?



Which is more efficient and less expensive?

*Check out the recording for the solution!*

# SER 321

## Serialization

What is it? 🤔

“Translating data structures or object states for storage or transmission”

Main Forms:

Text

or

Binary

?

- XML
- ➡ • JSON
- Java Serialization (Objects)
- ➡ • Protocol Buffers

*Check out the recording for the solution!*



# SER 321

## Serialization - JSON

Spitfire Review!

Data is put into...

"Katie"

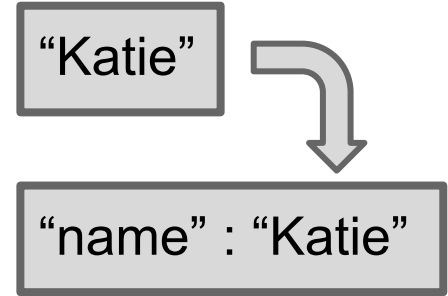
# SER 321

## Serialization - JSON

Data is put into...

Name : Value pairs

Spitfire Review!



# SER 321

## Serialization - JSON

Data is put into...

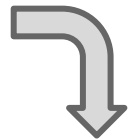
Name : Value pairs

AKA



Spitfire Review!

"Katie"



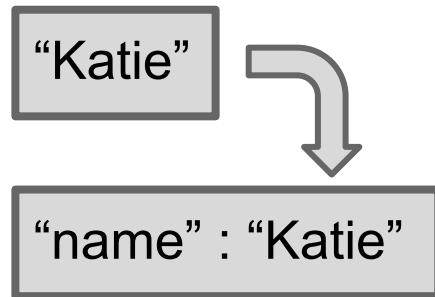
"name" : "Katie"

# SER 321

## Serialization - JSON

Spitfire Review!

Data is put into...



# SER 321

## Serialization - JSON

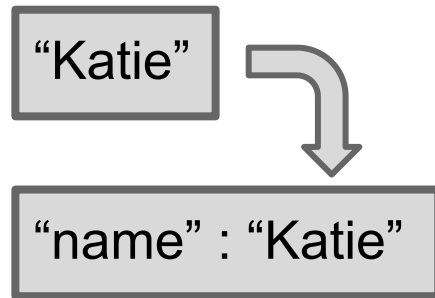
Spitfire Review!

Data is put into...



What uses curly braces?

{ }

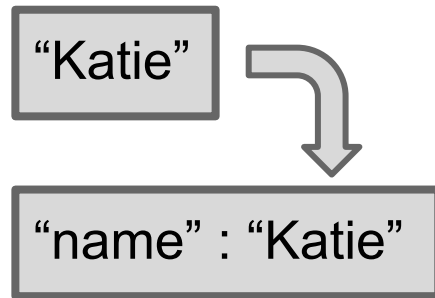


# SER 321

## Serialization - JSON

Spitfire Review!

Data is put into...



What uses curly braces?

Objects

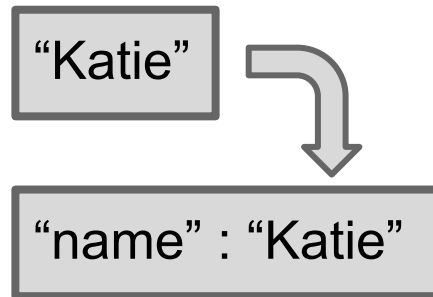
{ }

# SER 321

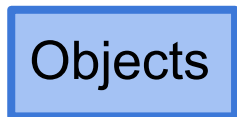
## Serialization - JSON

Spitfire Review!

Data is put into...

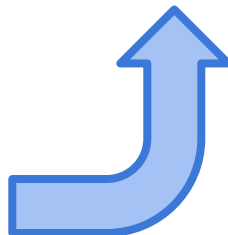


What uses curly braces?



{ }

What can objects hold?

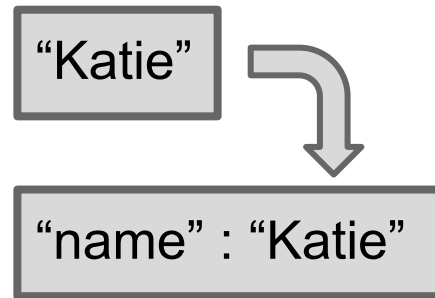


# SER 321

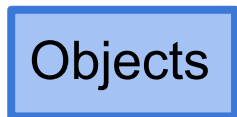
## Serialization - JSON

Spitfire Review!

Data is put into...

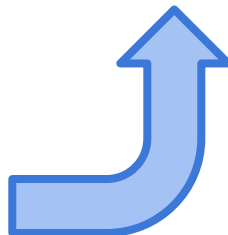
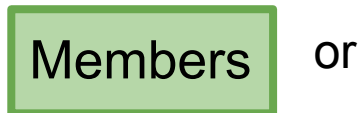


What uses curly braces?



{ }

What can objects hold?



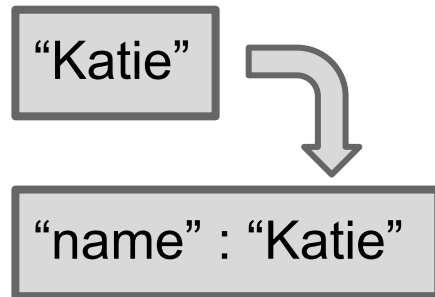


# SER 321

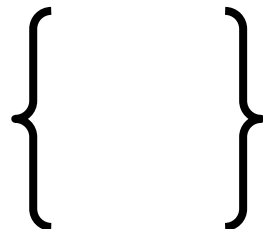
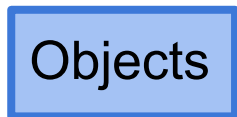
## Serialization - JSON

Spitfire Review!

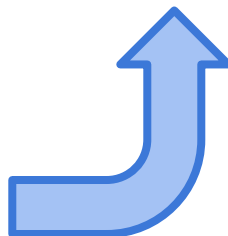
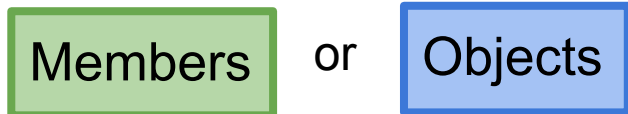
Data is put into...



What uses curly braces?



What can objects hold?



# SER 321

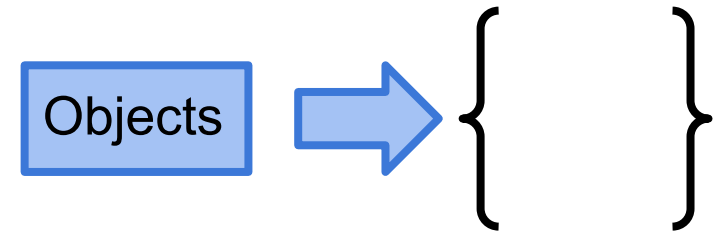
## Serialization - JSON

Spitfire Review!



What uses brackets?

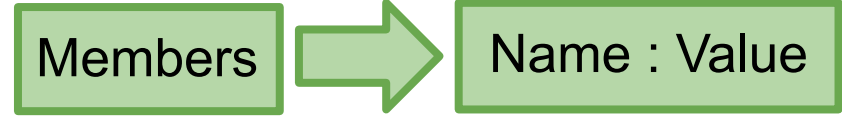
[ ]



# SER 321

## Serialization - JSON

Spitfire Review!



What uses brackets?

Arrays

[ ]

Objects

{ }

# SER 321

## Serialization - JSON

Spitfire Review!

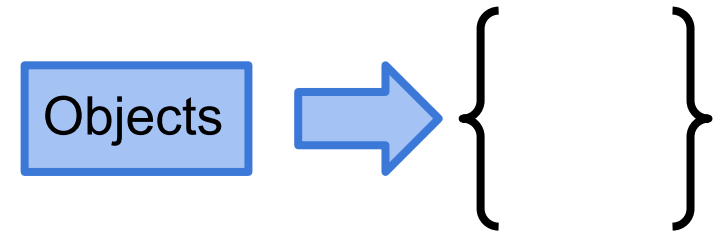
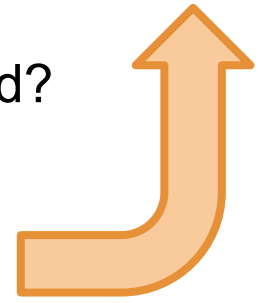


What uses brackets?

Arrays

[ ]

What can arrays hold?



# SER 321

## Serialization - JSON

Spitfire Review!

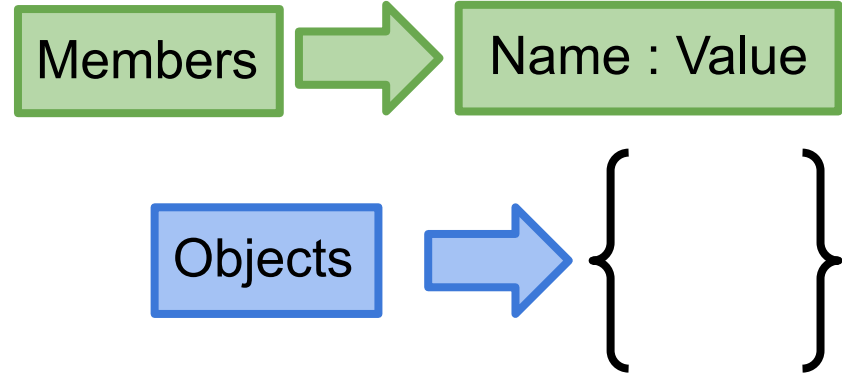
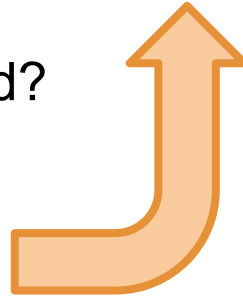
What uses brackets?

Arrays

[ ]

What can arrays hold?

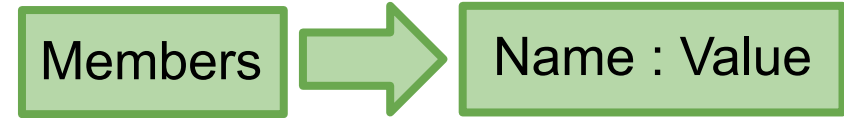
Any Valid Value



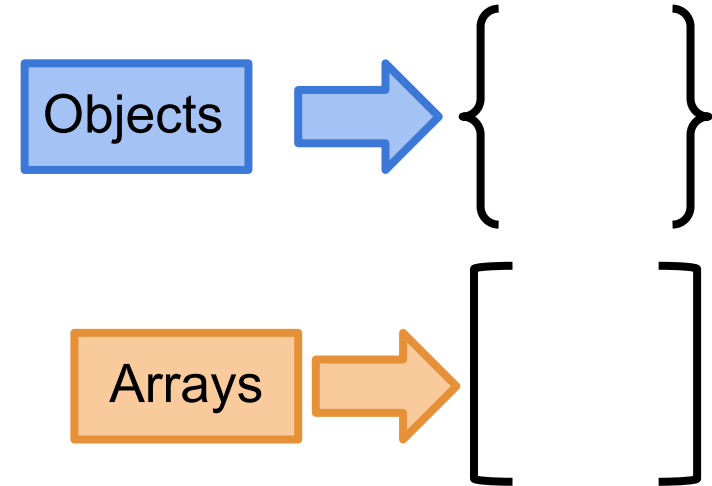
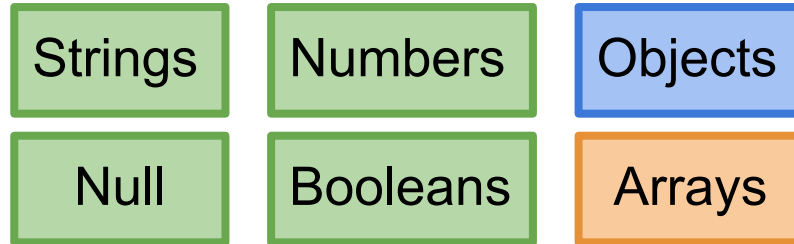
# SER 321

## Serialization - JSON

Spitfire Review!



What is a valid value?



Questions on JSON?

# SER 321

## Protocol Buffers

Require a few steps before use - listed in the README

Step 1: Generate the Protobuf Code

```
gradle generateProto
```

Step 2: If you use IntelliJ, add the following to your build.gradle file:

```
sourceSets {  
    main {  
        java {  
            srcDirs 'build/generated/source/proto/main/java'  
        }  
    }  
}
```

### Protocol Buffers

- *Message* is the base structure
- Valid messages are defined in a definition file (.proto)
- Protocol changes require updates to the definitions file
- .proto must be compiled before use
- Uses a **builder** to construct messages
- Handles the serialization for us!

- Serialize data for transmission
- Use IO streams

### JSON

- *Object* is the base structure
- JSON is valid if it adheres to the syntax rules we have learned
- Protocol changes result in logic modifications in both client and server
- No extra compilation needed
- Can use a library (*org.json*) to facilitate construction and use
- No additional serialization is done automatically



# SER 321

## Serialization - Protocol Buffers

How many message type definitions do we have here?

So what does a Response message need to contain?

Need to check the protocol!

## Protocol Elaboration

Check out the recording for the full walkthrough!

Starter Code for [Assign 4-2](#)

```
message Response {  
  enum ResponseType {  
    HELLO = 0;  
    LEADERBOARD = 1;  
    TASK = 2;  
    WON = 3;  
    ERROR = 4;  
    BYE = 5;  
  }  
  
  optional ResponseType responseType = 1 [default = HELLO];  
  optional string hello = 2;  
  
  repeated Leader leaderboard = 3;  
  
  optional string task = 4;  
  optional string image = 5;  
  
  optional bool eval = 6;  
  optional string message = 7;  
}  
  
message Leader {  
  optional string name = 1;  
  optional int32 wins = 2;  
  optional int32 logins = 3;  
}
```

# SER 321

## Serialization - Protocol Buffers


Let's do an error example...

So what does an Error Response message need to contain?

Need to check the protocol!

*Check out the recording for the full walkthrough!*

Starter Code for [Assign 4-2](#)



```
message Response {  
  enum ResponseType {  
    HELLO = 0;  
    LEADERBOARD = 1;  
    TASK = 2;  
    WON = 3;  
    ERROR = 4;  
    BYE = 5;  
  }  
  
  optional ResponseType responseType = 1 [default = HELLO];  
  optional string hello = 2;  
  
  repeated Leader leaderboard = 3;  
  
  optional string task = 4;  
  optional string image = 5;  
  
  optional bool eval = 6;  
  optional string message = 7;  
}  
  
message Leader {  
  optional string name = 1;  
  optional int32 wins = 2;  
  optional int32 logins = 3;  
}
```

# SER 321

## Serialization - Protocol Buffers

Let's do an error example...

So what does an Error Response message need to contain?

```
SV Response
```

```
...
```

```
ResponseType: ERROR
```

```
RequiredFields: message (description of error), type
```

```
...
```

```
Some error types to use:
```

```
1 - required field missing -- in message name the field
```

```
2 - request not supported -- in message name the request that is not supported
```

```
0 - any other errors, in this case the message will just be displayed
```

```
message Response {  
  enum ResponseType {  
    HELLO = 0;  
    LEADERBOARD = 1;  
    TASK = 2;  
    WON = 3;  
    ERROR = 4;  
    BYE = 5;  
  }  
}
```

```
al ResponseType responseType = 1 [default = HELLO];  
al string hello = 2;
```

```
ed Leader leaderboard = 3;
```

```
al string task = 4;
```

```
al string image = 5;
```

```
al bool eval = 6;
```

```
al string message = 7;
```

```
}
```

```
message Leader {  
  optional string name = 1;  
  optional int32 wins = 2;  
  optional int32 logins = 3;  
}
```

*Check out the recording for the full walkthrough!*

Starter Code for [Assign 4-2](#)

# SER 321

## Serialization - Protocol Buffers

ResponseType

Message



Let's do an error example...

So what does an Error Response message need to contain?

```
SV Response
```

```
...
```

```
ResponseType: ERROR
```

```
RequiredFields: message (description of error), type
```

```
...
```

```
Some error types to use:
```

```
1 - required field missing -- in message name the field
```

```
2 - request not supported -- in message name the request that is not supported
```

```
0 - any other errors, in this case the message will just be displayed
```

```
message Response {  
  enum ResponseType {  
    HELLO = 0;  
    LEADERBOARD = 1;  
    TASK = 2;  
    WON = 3;  
    ERROR = 4;  
    BYE = 5;  
  }  
}
```

```
al ResponseType responseType = 1 [default = HELLO];
```

```
al string hello = 2;
```

```
ed Leader leaderboard = 3;
```

```
al string task = 4;
```

```
al string image = 5;
```

```
al bool eval = 6;
```

```
al string message = 7;
```

```
}
```

```
message Leader {  
  optional string name = 1;  
  optional int32 wins = 2;  
  optional int32 logins = 3;  
}
```

*Check out the recording for the full walkthrough!*

Starter Code for [Assign 4-2](#)

# SER 321

## Serialization - Protocol Buffers

ResponseType

Message



Let's do an error example...

So what does an Error Response message need to contain?

SV Response

...

ResponseType: ERROR

RequiredFields: message (description of error), type

...

Some error types to use:

1 - required field missing -- in message name the field

2 - request not supported -- in message name the request that is not supported

0 - any other errors, in this case the message will just be displayed

```
message Response {  
  enum ResponseType {  
    HELLO = 0;  
    LEADERBOARD = 1;  
    TASK = 2;  
    WON = 3;  
    ERROR = 4;  
    BYE = 5;  
  }  
}
```

```
al ResponseType responseType = 1 [default = HELLO];  
al string hello = 2;
```

```
ed Leader leaderboard = 3;
```

```
al string task = 4;
```

```
al string image = 5;
```

```
al bool eval = 6;
```

```
al string message = 7;
```

```
}
```

```
message Leader {  
  optional string name = 1;  
  optional int32 wins = 2;  
  optional int32 logins = 3;  
}
```

```
}
```

Check out the recording for the full walkthrough!

Starter Code for [Assign 4-2](#)

# SER 321

## Serialization - Protocol Buffers

How would we send the first task to the client?

```
SV Response
Server responds with a message specifying if the game is joined or started.
...
ResponseType: TASK
Required Fields: image, task
...
```

*Check out the recording for the full walkthrough!*

Starter Code for [Assign 4-2](#)

```
message Response {
  enum ResponseType {
    HELLO = 0;
    LEADERBOARD = 1;
    TASK = 2;
    WON = 3;
    ERROR = 4;
    BYE = 5;
  }

  optional ResponseType responseType = 1 [default = HELLO];
  optional string hello = 2;

  repeated Leader leaderboard = 3;

  optional string task = 4;
  optional string image = 5;

  optional bool eval = 6;
  optional string message = 7;
}

message Leader {
  optional string name = 1;
  optional int32 wins = 2;
  optional int32 logins = 3;
}
```

# SER 321

## Serialization - Protocol Buffers

What if I need to construct the message slowly?

Use Response.Builder much like a StringBuilder!

```
Response response2 = Response.newBuilder()  
    .setResponseType(Response.ResponseType.TASK)  
    .setImage(game.getImage())  
    .setTask("Great task goes here")  
    .build(); ★
```

```
Leader leader2 = Leader.newBuilder()  
    .setName("name2")  
    .setWins(1)  
    .setLogins(1)  
    .build(); ★
```



Is that it?

*Check out the recording for the full walkthrough!*

```
Response.Builder res = Response.newBuilder()  
    .setResponseType(Response.ResponseType.LEADERBOARD);
```

```
res.addLeaderboard(leader);  
res.addLeaderboard(leader2);
```

Starter Code for [Assign 4-2](#)

```
response3.writeDelimitedTo(out);
```



# Questions?

## Survey:

[https://bit.ly/asn\\_survey](https://bit.ly/asn_survey)





## Upcoming Events

### SI Sessions:

- Thursday, November 9th 2023 at 7:00 pm MST
- Sunday, November 12th 2023 at 7:00 pm MST
- Monday, November 13th 2023 at 4:00 pm MST

### Review Sessions:

- Finishing up the Scheduling Poll today, will post tonight or tomorrow!

# 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!

## Additional Resources

[CoureRepo](#)

[org.json API Docs](#)

[JSON Helper](#)

[Dining Philosophers Interactive](#)

[Austin Walter's Traffic Comparison](#)