SER 321 B Session

SI Session

Thursday, March 21st 2024

7:00 pm - 8:00 pm MST

Agenda

Gradle

JSON

Making your Code Robust

Client & Server

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

Helper Document is ***super*** helpful!

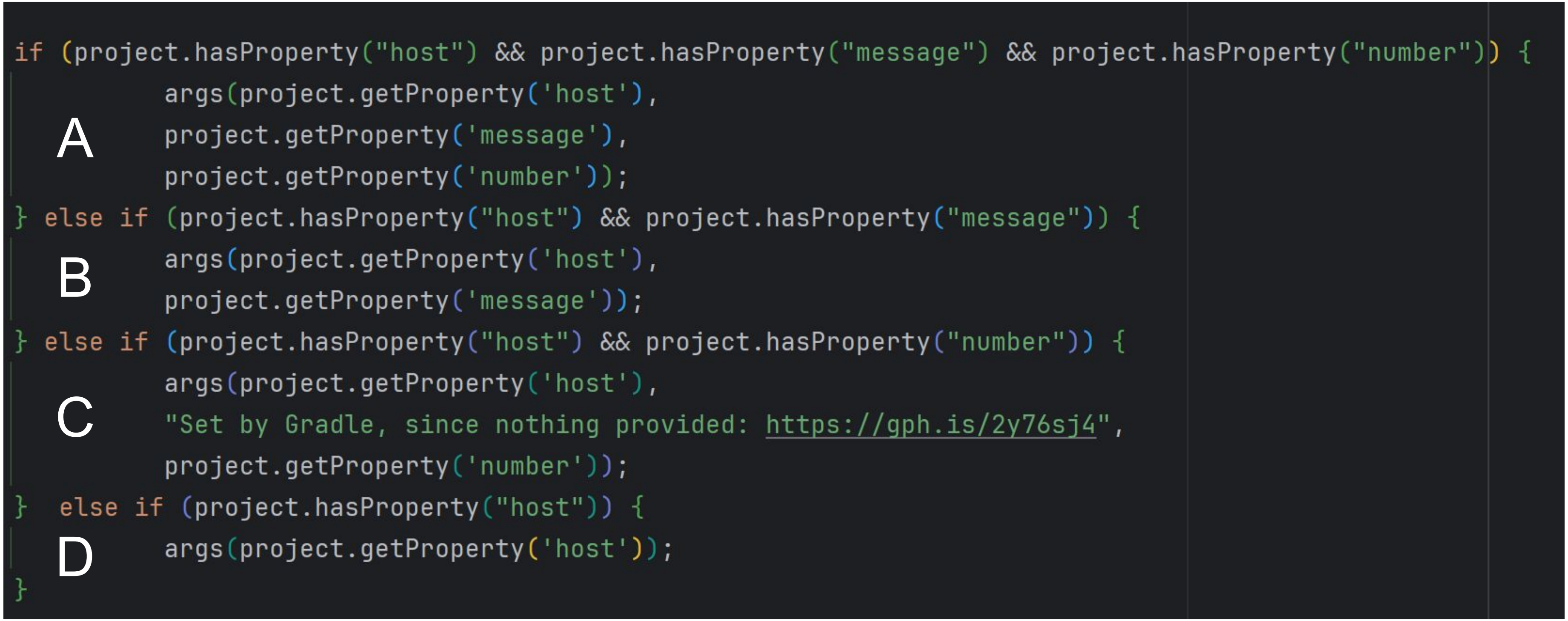
👏Thank you UGTA Paul Heaton! 👏

Sockets/JavaSimpleSock2

SER 321
Gradle Review

Which statement will execute with…

```
gradle SocketClient -Phost=localhost -Pmessage=secret
```



```
if (project.hasProperty("host") && project.hasProperty("message") && project.hasProperty("number")) {
        args(project.getProperty('host'),
        project.getProperty('message'),
        project.getProperty('number'));
} else if (project.hasProperty("host") && project.hasProperty("message")) {
        args(project.getProperty('host'),
        project.getProperty('message'));
} else if (project.hasProperty("host") && project.hasProperty("number")) {
        args(project.getProperty('host'),
        "Set by Gradle, since nothing provided: https://gph.is/2y76sj4",
        project.getProperty('number'));
}  else if (project.hasProperty("host")) {
        args(project.getProperty('host'));
}
```

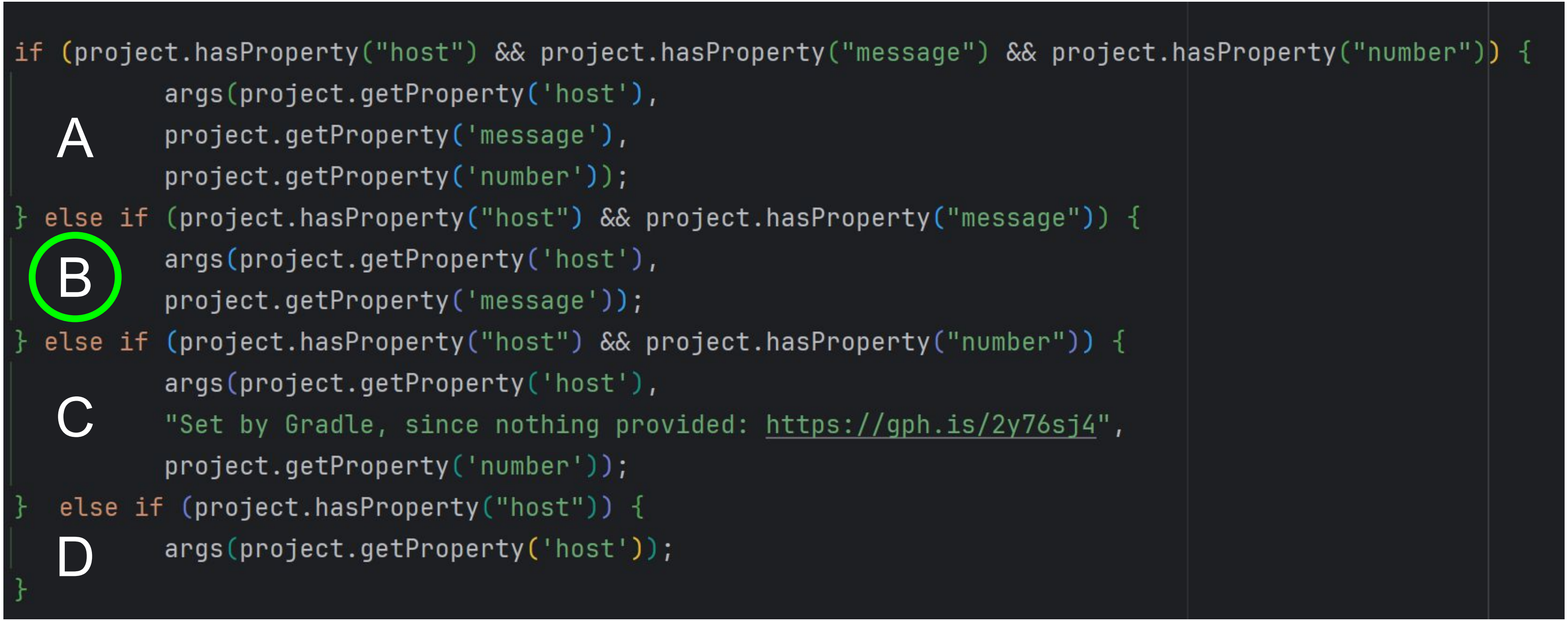
Sockets/JavaSimpleSock2

SER 321

Gradle Review

Which statement will execute with…

```
gradle SocketClient -Phost=localhost -Pmessage=secret
```



```
if (project.hasProperty("host") && project.hasProperty("message") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          project.getProperty('message'),
          project.getProperty('number'));
} else if (project.hasProperty("host") && project.hasProperty("message")) {
          args(project.getProperty('host'),
          project.getProperty('message'));
} else if (project.hasProperty("host") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          "Set by Gradle, since nothing provided: https://gph.is/2y76sj4",
          project.getProperty('number'));
}  else if (project.hasProperty("host")) {
          args(project.getProperty('host'));
}
```

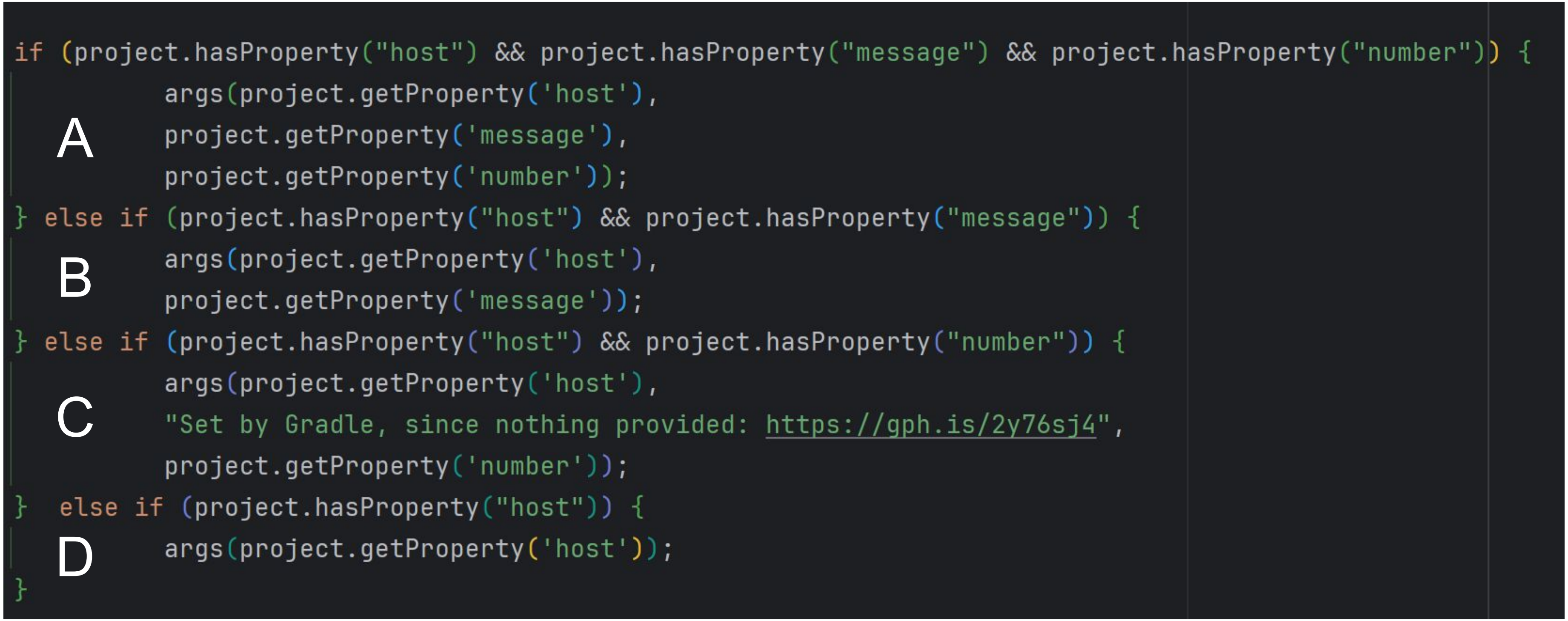
Sockets/JavaSimpleSock2

SER 321

Gradle Review

Which statement will execute with…

```
gradle SocketClient -Phost=localhost -Pmessage=secret -Pnumber=2
```



```
if (project.hasProperty("host") && project.hasProperty("message") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          project.getProperty('message'),
          project.getProperty('number'));
} else if (project.hasProperty("host") && project.hasProperty("message")) {
          args(project.getProperty('host'),
          project.getProperty('message'));
} else if (project.hasProperty("host") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          "Set by Gradle, since nothing provided: https://gph.is/2y76sj4",
          project.getProperty('number'));
}  else if (project.hasProperty("host")) {
          args(project.getProperty('host'));
}
```

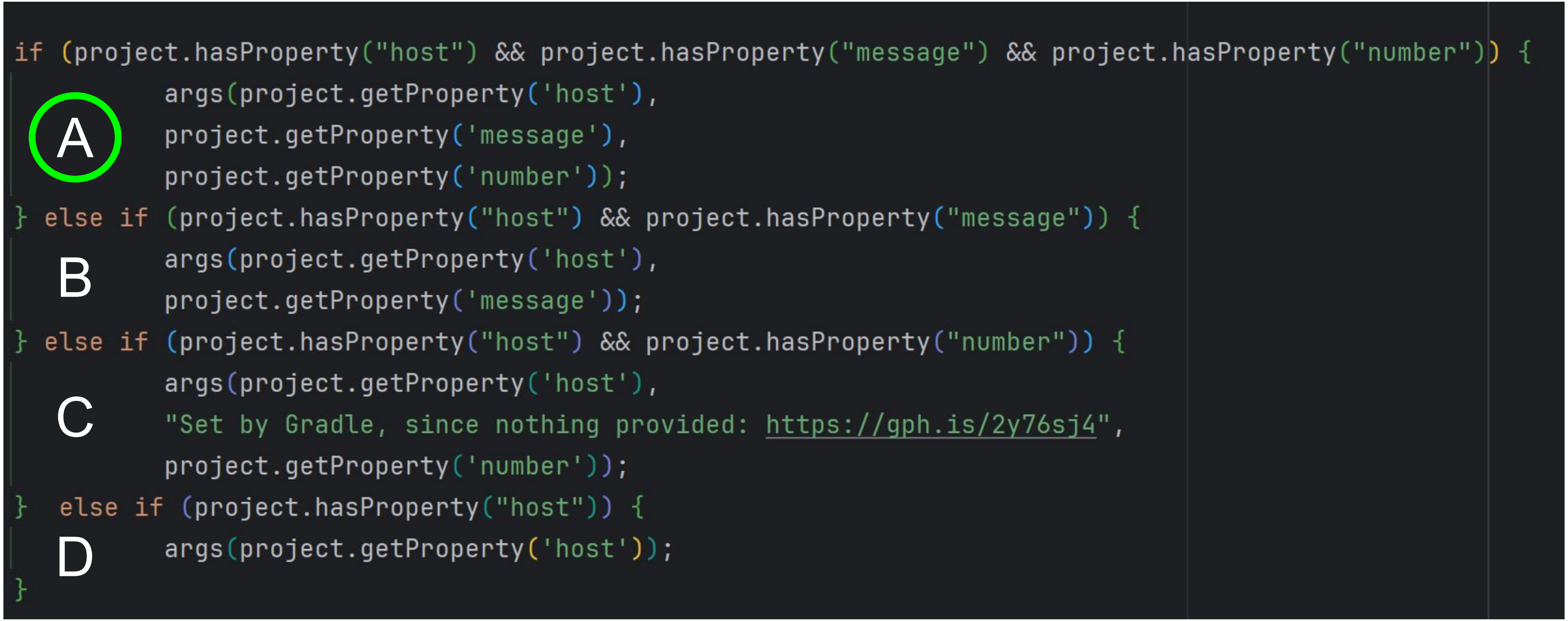
Sockets/JavaSimpleSock2

SER 321

Gradle Review

Which statement will execute with…

```
gradle SocketClient -Phost=localhost -Pmessage=secret -Pnumber=2
```



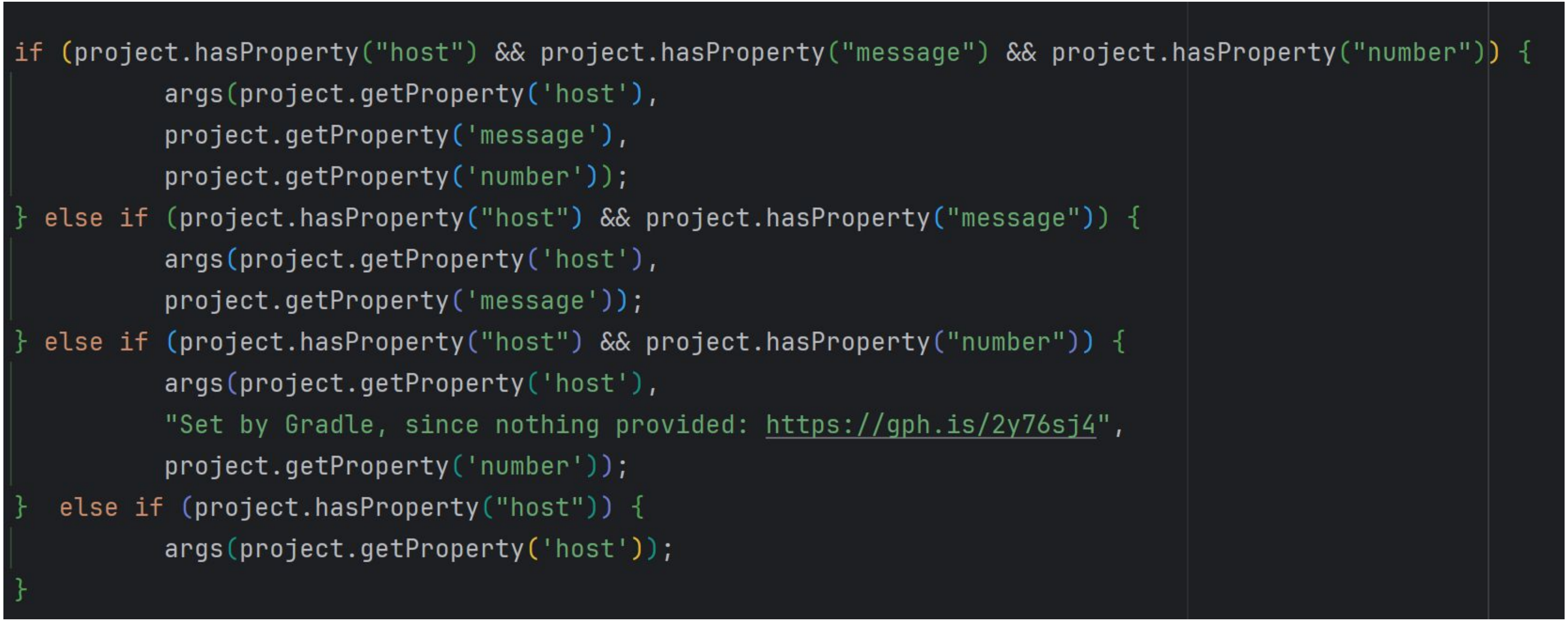
```
if (project.hasProperty("host") && project.hasProperty("message") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          project.getProperty('message'),
          project.getProperty('number'));
} else if (project.hasProperty("host") && project.hasProperty("message")) {
          args(project.getProperty('host'),
          project.getProperty('message'));
} else if (project.hasProperty("host") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          "Set by Gradle, since nothing provided: https://gph.is/2y76sj4",
          project.getProperty('number'));
}  else if (project.hasProperty("host")) {
          args(project.getProperty('host'));
}
```

Sockets/JavaSimpleSock2

SER 321

Gradle Review

Have we set any defaults here?

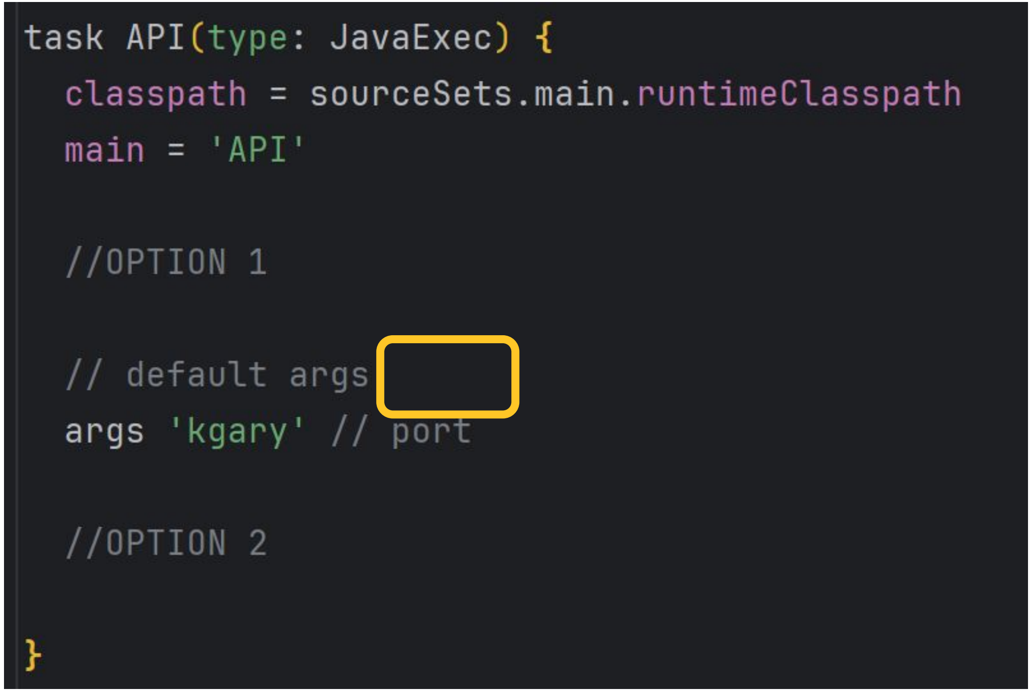


```
if (project.hasProperty("host") && project.hasProperty("message") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          project.getProperty('message'),
          project.getProperty('number'));
} else if (project.hasProperty("host") && project.hasProperty("message")) {
          args(project.getProperty('host'),
          project.getProperty('message'));
} else if (project.hasProperty("host") && project.hasProperty("number")) {
          args(project.getProperty('host'),
          "Set by Gradle, since nothing provided: https://gph.is/2y76sj4",
          project.getProperty('number'));
}  else if (project.hasProperty("host")) {
          args(project.getProperty('host'));
}
```

SER 321

Gradle Review

Let’s modify this code to have *both* defaults and command line properties!



```
task API(type: JavaExec) {
  classpath = sourceSets.main.runtimeClasspath
  main = 'API'

  //OPTION 1

  // default args
  args 'kgary' // port

  //OPTION 2

}
```

First let’s make the *if* statement:

Okay, now where do we put it?

org.json Docs

JSON Guide

SER 321

JSON

Do you see any errors?

```
{
        “name” : “katie”,
        “role” : “student”,
        “course” : “ser321”
}
```

```
[{
        “name” : “katie”,
        “name” : “zac”
}]
```

```
{
        “name” : “katie”,
        “pets” : {
                “dog” : “smokey”,
                “dog” : “samie”
                }
}
```

```
{
        “submissions” : [
                {“name” : “katie”},
                {“name” : “zac”}
        ]
}
```

SER 321

JSON

org.json Docs

How could we correct them?

JSON Guide

```
[
        {“name” : “katie”},
        {“name” : “zac”}
]
```

```
[{
        “name” : “katie”,
        “name” : “zac”
}]
```

```
{
        “name” : “katie”,
        “pets” : {
                “dog” : “smokey”,
                “dog” : “samie”
                }
}
```

```
{
        “name” : “katie”,
        “pets” : {
                {“dog” : “smokey”},
                {“dog” : “samie”}
                }
}
```

SER 321

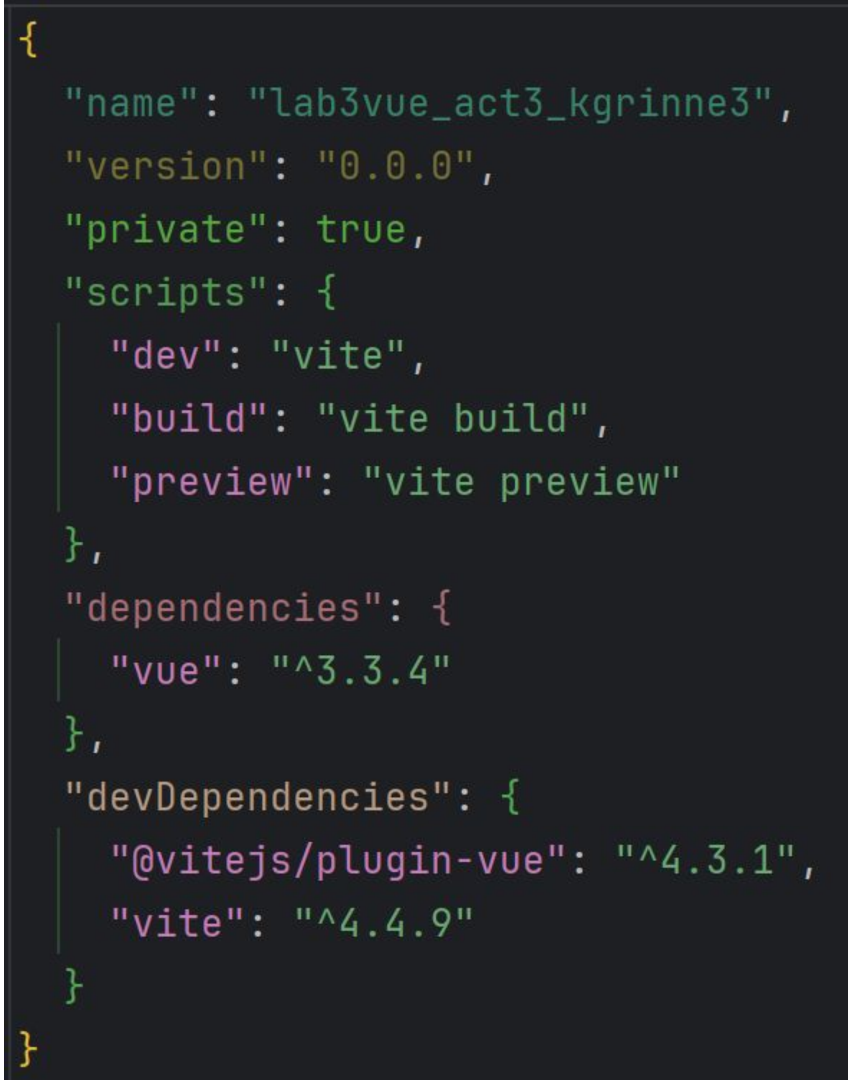
JSON

org.json Docs

How many Objects?

How many Arrays?

How many Members?



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

JSON Guide

org.json Docs

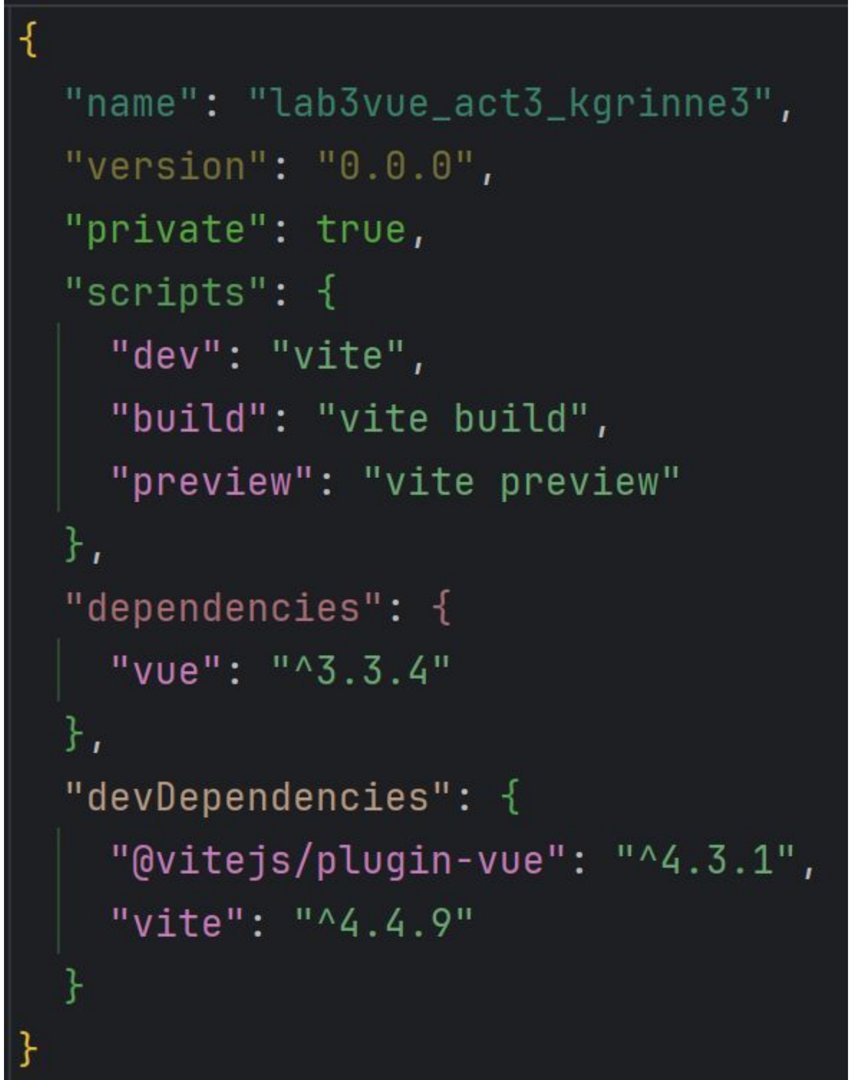
SER 321

JSON

```
JSONObject json =
```

What if I want to see if name exists first?

How about the dev value?



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

JSON Guide

SER 321

Making your Code Robust

What do we mean when we say “make sure your code is robust”?

Error Handling

SER 321

Making your Code Robust

What do we mean when we say “make sure your code is robust”?

You

Buddy

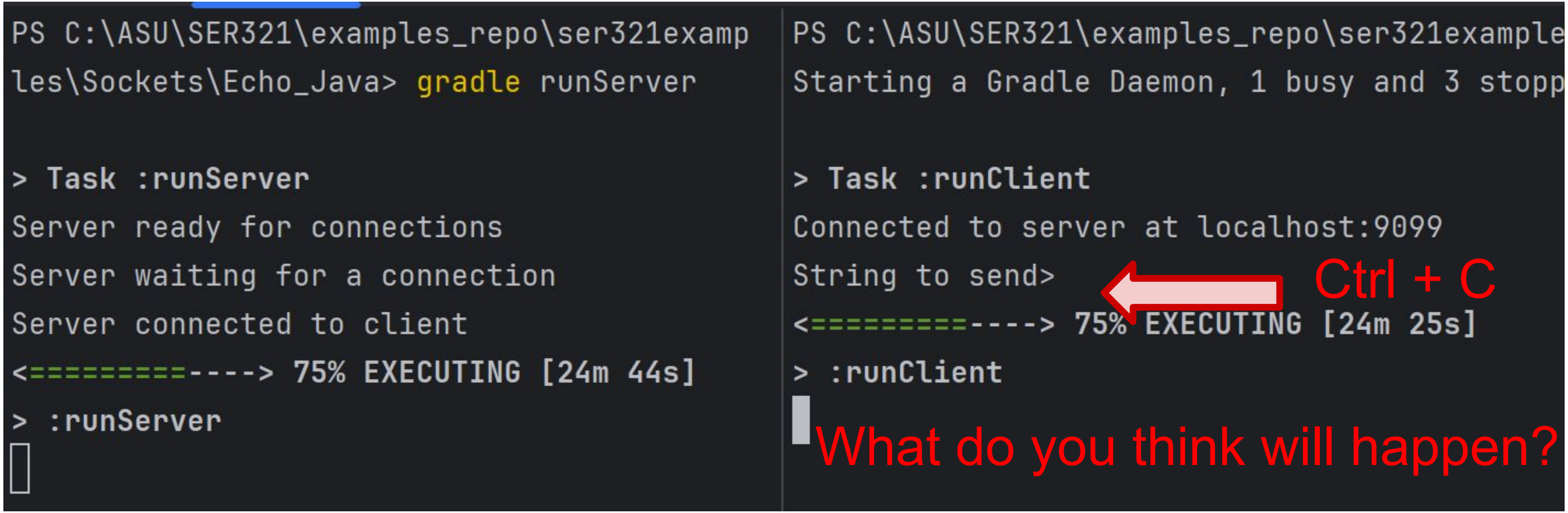
```
An error has occurred. To continue:
                                        Windows
Press Enter to return to Windows, or
Press CTRL+ALT+DEL to restart your computer. If you do this,
you will lose any unsaved information in all open applications.
Error: 0E : 016F : BFF9B3D4

              Press any key to continue _
```

Sockets/Echo_Java

SER 321

Making your Code Robust

What do we mean when we say “make sure your code is robust”?



```
PS C:\ASU\SER321\examples_repo\ser321examp
les\Sockets\Echo_Java> gradle runServer

> Task :runServer
Server ready for connections
Server waiting for a connection
Server connected to client
<=========----> 75% EXECUTING [24m 44s]
> :runServer
```

```
PS C:\ASU\SER321\examples_repo\ser321example
Starting a Gradle Daemon, 1 busy and 3 stopp

> Task :runClient
Connected to server at localhost:9099
String to send>
<=========----> 75% EXECUTING [24m 25s]
> :runClient
```

Ctrl + C

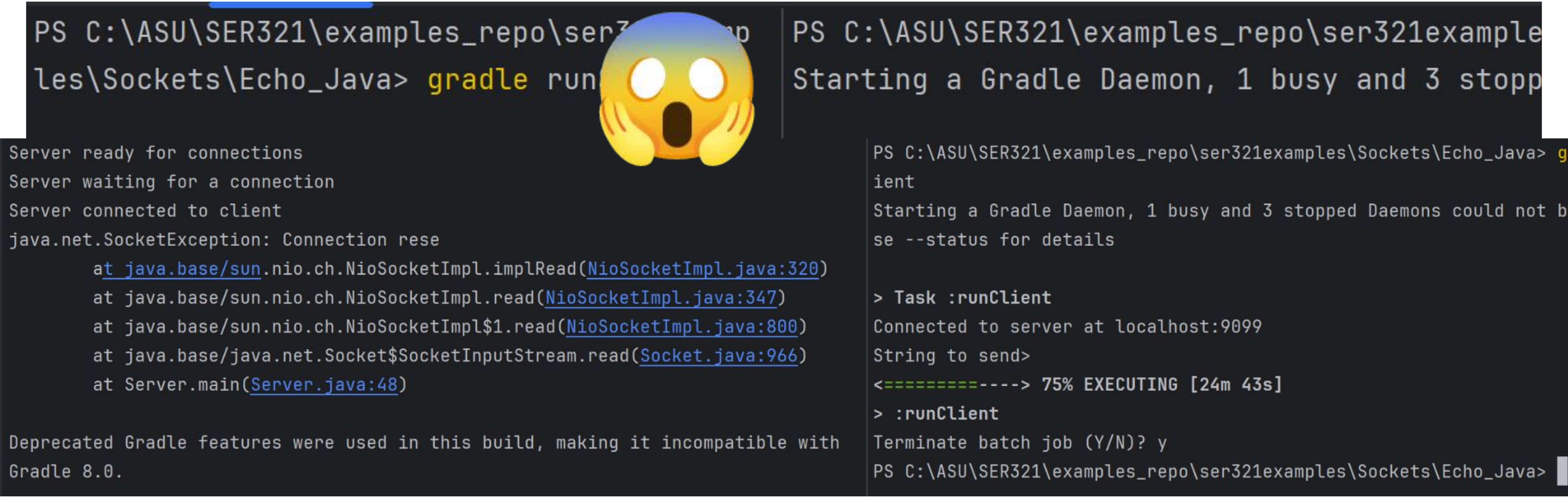
What do you think will happen?

Sockets/Echo_Java

SER 321

Making your Code Robust

We crashed the server!



```
PS C:\ASU\SER321\examples_repo\ser3        p
les\Sockets\Echo_Java> gradle run
```

```
PS C:\ASU\SER321\examples_repo\ser321example
Starting a Gradle Daemon, 1 busy and 3 stopp
```

```
Server ready for connections
Server waiting for a connection
Server connected to client
java.net.SocketException: Connection rese
        at java.base/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:320)
        at java.base/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:347)
        at java.base/sun.nio.ch.NioSocketImpl$1.read(NioSocketImpl.java:800)
        at java.base/java.net.Socket$SocketInputStream.read(Socket.java:966)
        at Server.main(Server.java:48)

Deprecated Gradle features were used in this build, making it incompatible with
Gradle 8.0.
```

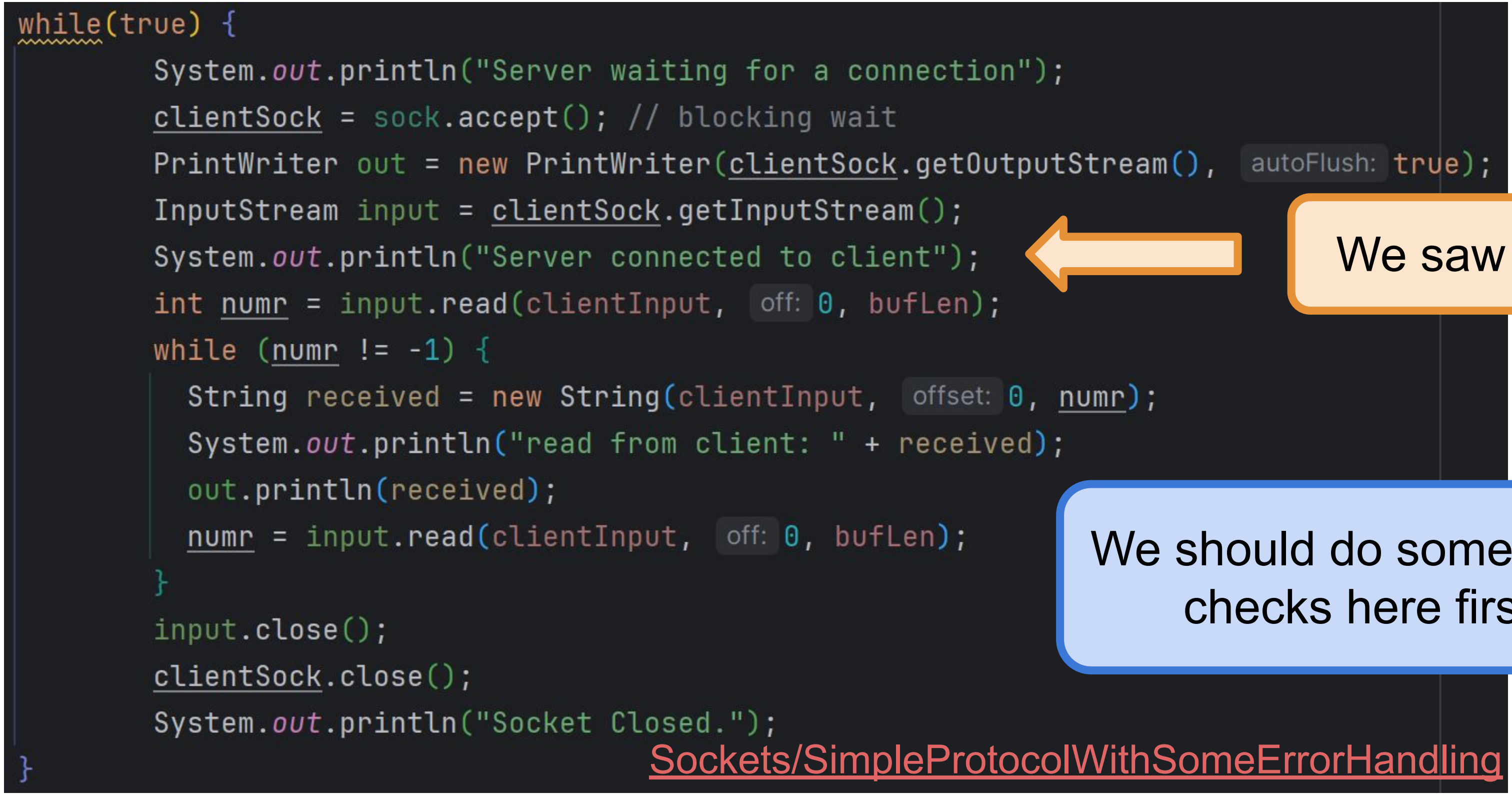
```
PS C:\ASU\SER321\examples_repo\ser321examples\Sockets\Echo_Java> g
ient
Starting a Gradle Daemon, 1 busy and 3 stopped Daemons could not b
se --status for details

> Task :runClient
Connected to server at localhost:9099
String to send>
<=========----> 75% EXECUTING [24m 43s]
> :runClient
Terminate batch job (Y/N)? y
PS C:\ASU\SER321\examples_repo\ser321examples\Sockets\Echo_Java>
```

SER 321

Making your Code Robust

What happened?

Sockets/Echo_Java



```java
while(true) {
        System.out.println("Server waiting for a connection");
        clientSock = sock.accept(); // blocking wait
        PrintWriter out = new PrintWriter(clientSock.getOutputStream(), autoFlush: true);
        InputStream input = clientSock.getInputStream();
        System.out.println("Server connected to client");
        int numr = input.read(clientInput, off: 0, bufLen);
        while (numr != -1) {
          String received = new String(clientInput, offset: 0, numr);
          System.out.println("read from client: " + received);
          out.println(received);
          numr = input.read(clientInput, off: 0, bufLen);
        }
        input.close();
        clientSock.close();
        System.out.println("Socket Closed.");
}
```

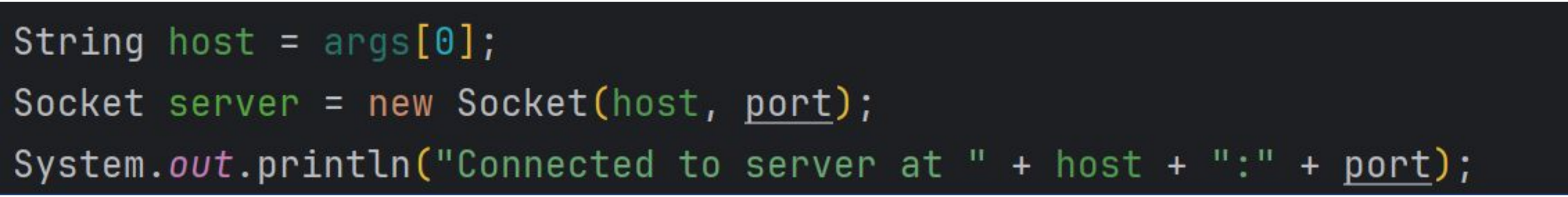
We saw this…

We should do some basic checks here first!

Sockets/SimpleProtocolWithSomeErrorHandling

Sockets/Echo_Java

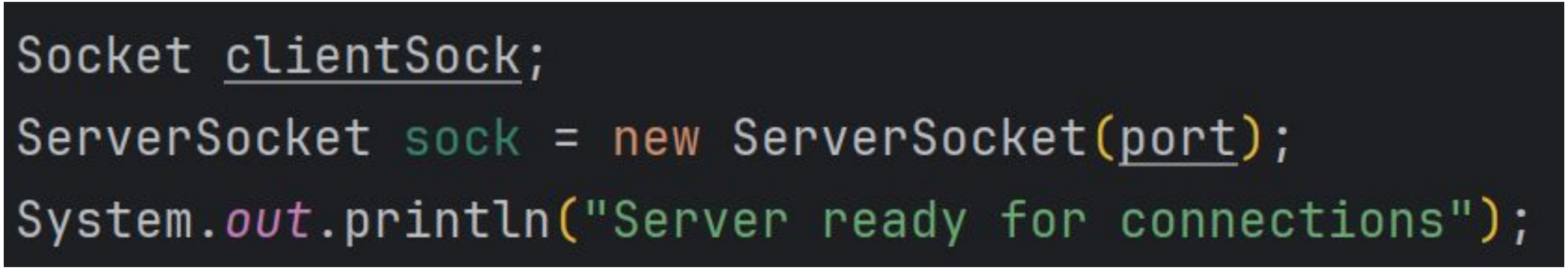
Challenge - Client or Server?



```
String host = args[0];
Socket server = new Socket(host, port);
System.out.println("Connected to server at " + host + ":" + port);
```

Sockets/Echo_Java

Challenge - Client or Server?



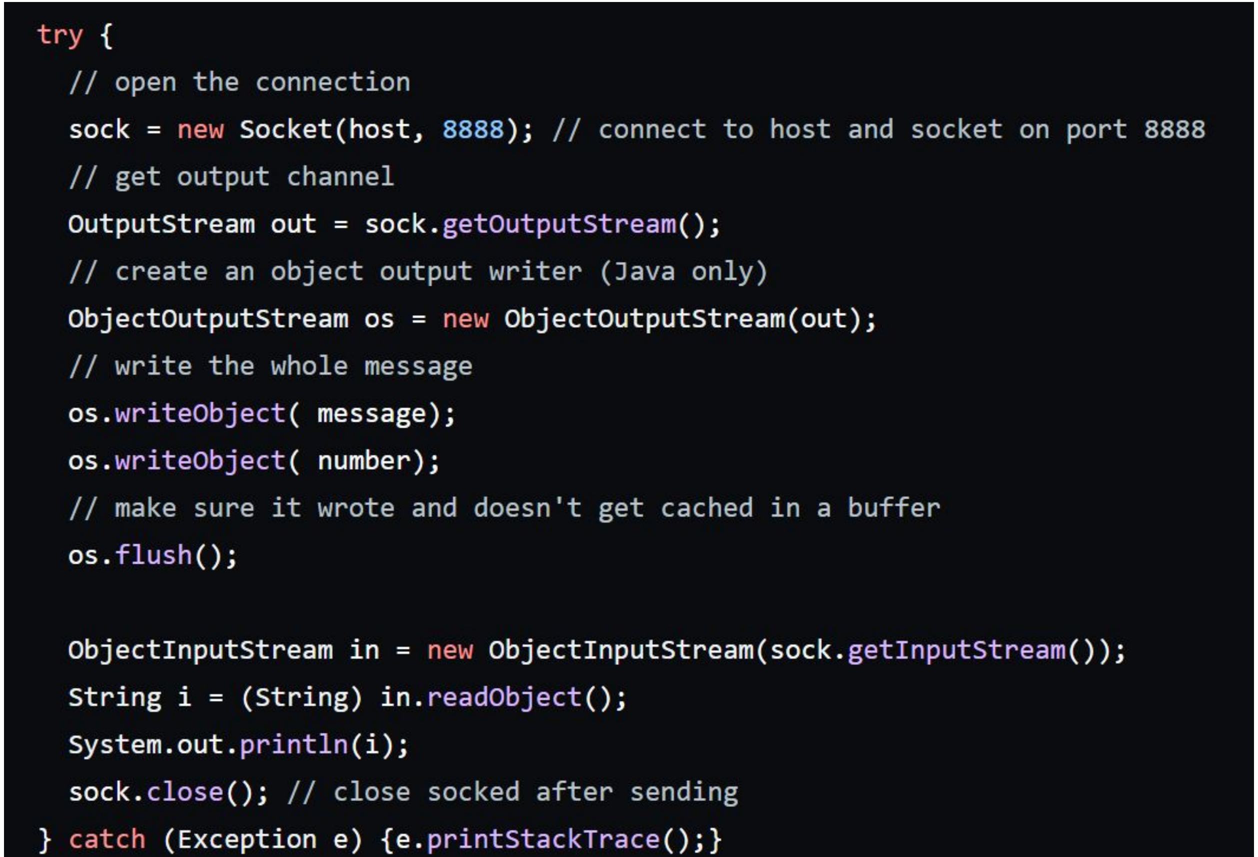
```
Socket clientSock;
ServerSocket sock = new ServerSocket(port);
System.out.println("Server ready for connections");
```

How are we feeling about Client/Server communications so far?

Sockets/JavaSimpleSock2

SER 321

Client/Server

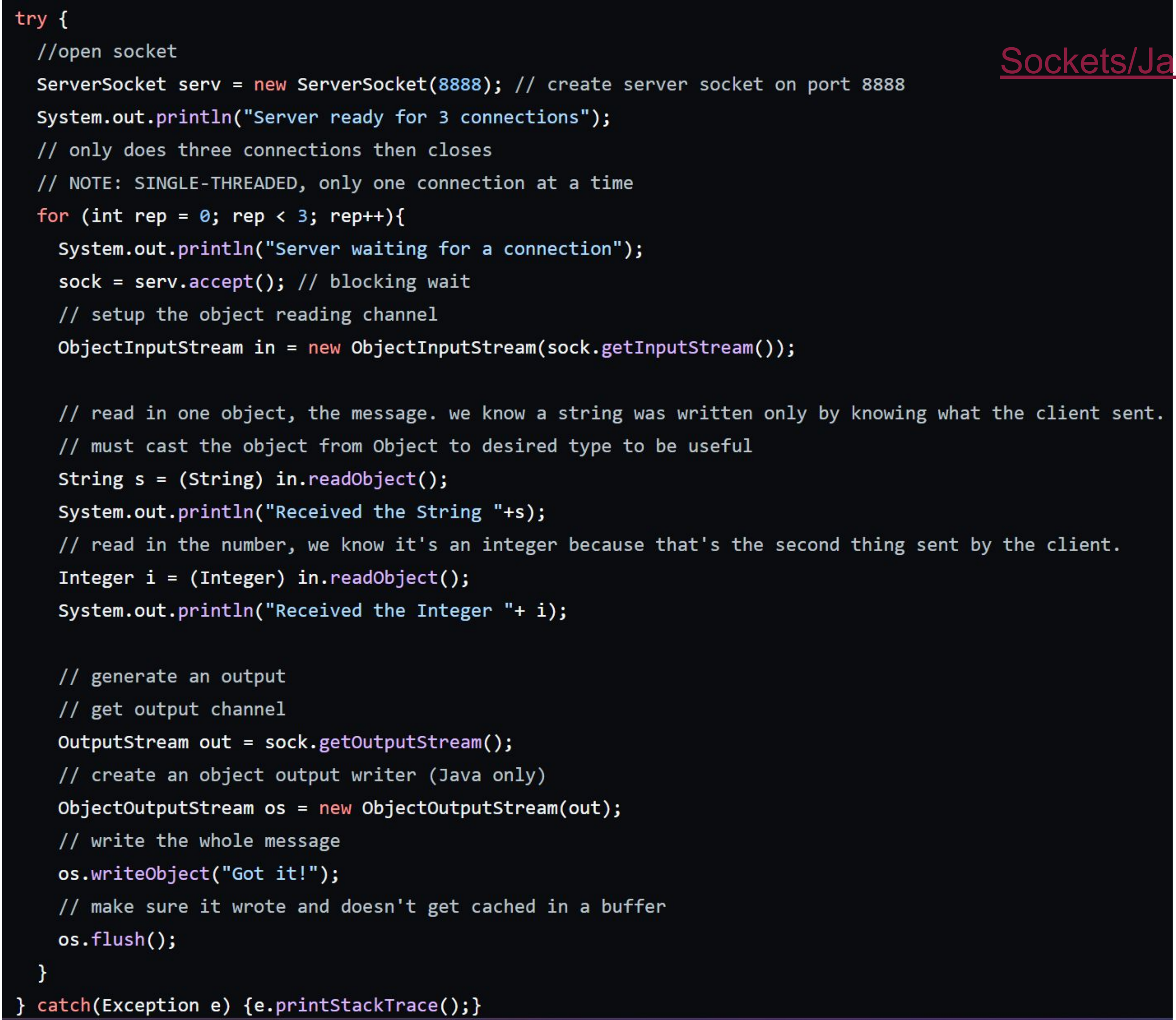


```
try {
  // open the connection
  sock = new Socket(host, 8888); // connect to host and socket on port 8888
  // get output channel
  OutputStream out = sock.getOutputStream();
  // create an object output writer (Java only)
  ObjectOutputStream os = new ObjectOutputStream(out);
  // write the whole message
  os.writeObject( message);
  os.writeObject( number);
  // make sure it wrote and doesn't get cached in a buffer
  os.flush();

  ObjectInputStream in = new ObjectInputStream(sock.getInputStream());
  String i = (String) in.readObject();
  System.out.println(i);
  sock.close(); // close socked after sending
} catch (Exception e) {e.printStackTrace();}
```

SER 321

Client/Server

Sockets/JavaSimpleSock2



```java
try {
  //open socket
  ServerSocket serv = new ServerSocket(8888); // create server socket on port 8888
  System.out.println("Server ready for 3 connections");
  // only does three connections then closes
  // NOTE: SINGLE-THREADED, only one connection at a time
  for (int rep = 0; rep < 3; rep++){
    System.out.println("Server waiting for a connection");
    sock = serv.accept(); // blocking wait
    // setup the object reading channel
    ObjectInputStream in = new ObjectInputStream(sock.getInputStream());

    // read in one object, the message. we know a string was written only by knowing what the client sent.
    // must cast the object from Object to desired type to be useful
    String s = (String) in.readObject();
    System.out.println("Received the String "+s);
    // read in the number, we know it's an integer because that's the second thing sent by the client.
    Integer i = (Integer) in.readObject();
    System.out.println("Received the Integer "+ i);

    // generate an output
    // get output channel
    OutputStream out = sock.getOutputStream();
    // create an object output writer (Java only)
    ObjectOutputStream os = new ObjectOutputStream(out);
    // write the whole message
    os.writeObject("Got it!");
    // make sure it wrote and doesn't get cached in a buffer
    os.flush();
  }
} catch(Exception e) {e.printStackTrace();}
```

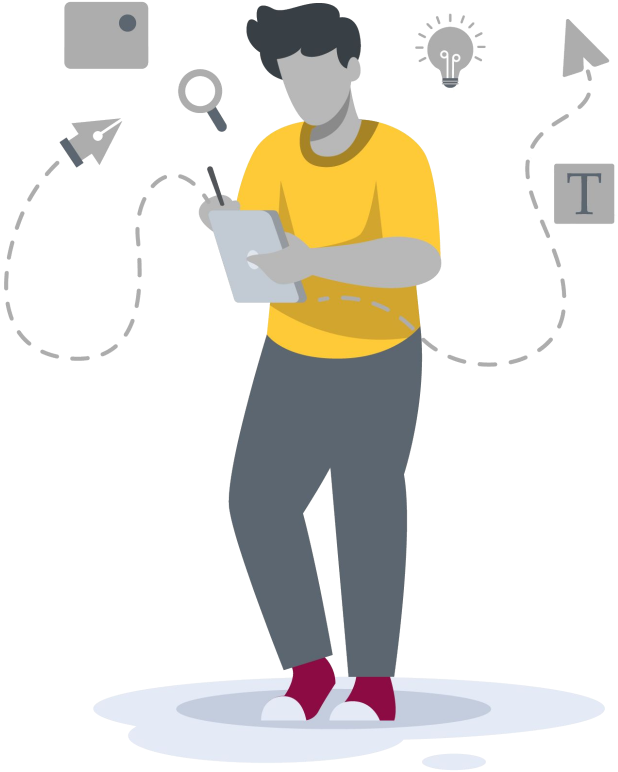
SER 321

Scratch Space

Questions?

Survey:

http://bit.ly/ASN2324



Upcoming Events

SI Sessions:

- Sunday, March 24th at 7:00 pm MST
- Monday, March 25th at 7:00 pm MST
- Thursday, March 28th at 7:00 pm MST

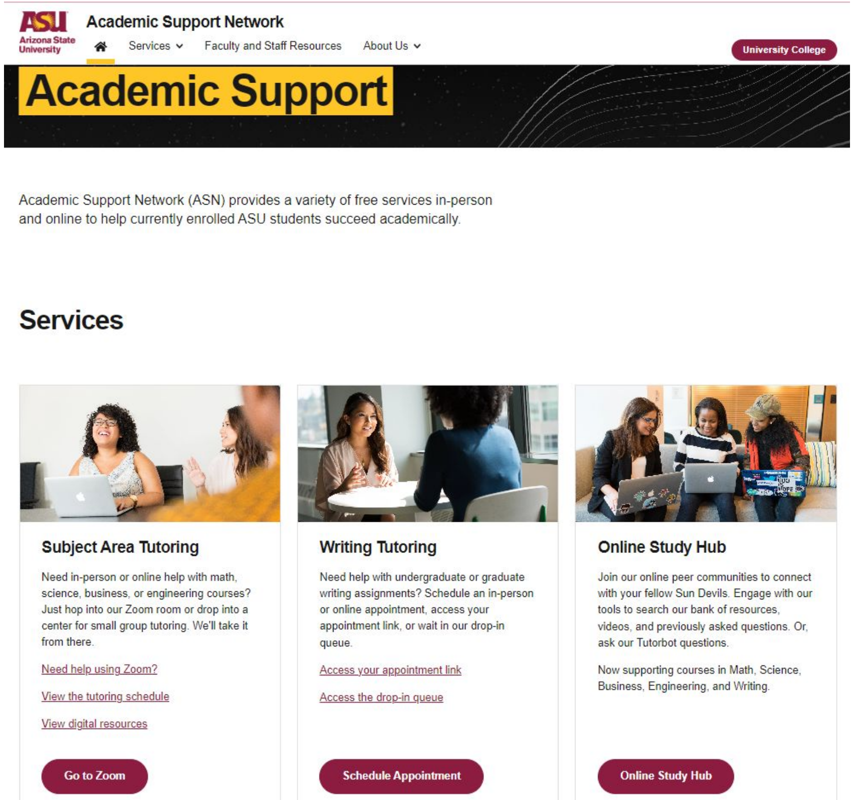
Review Sessions:

- TBD

More Questions?

Check out our other resources!

tutoring.asu.edu



ASU Arizona State University

Academic Support Network

Services ⌄   Faculty and Staff Resources   About Us ⌄

University College

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

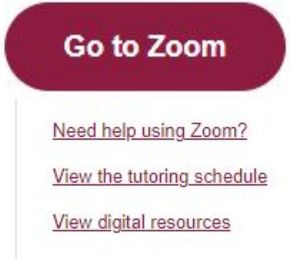
Schedule Appointment

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

Need help using Zoom?

2-

View the tutoring schedule

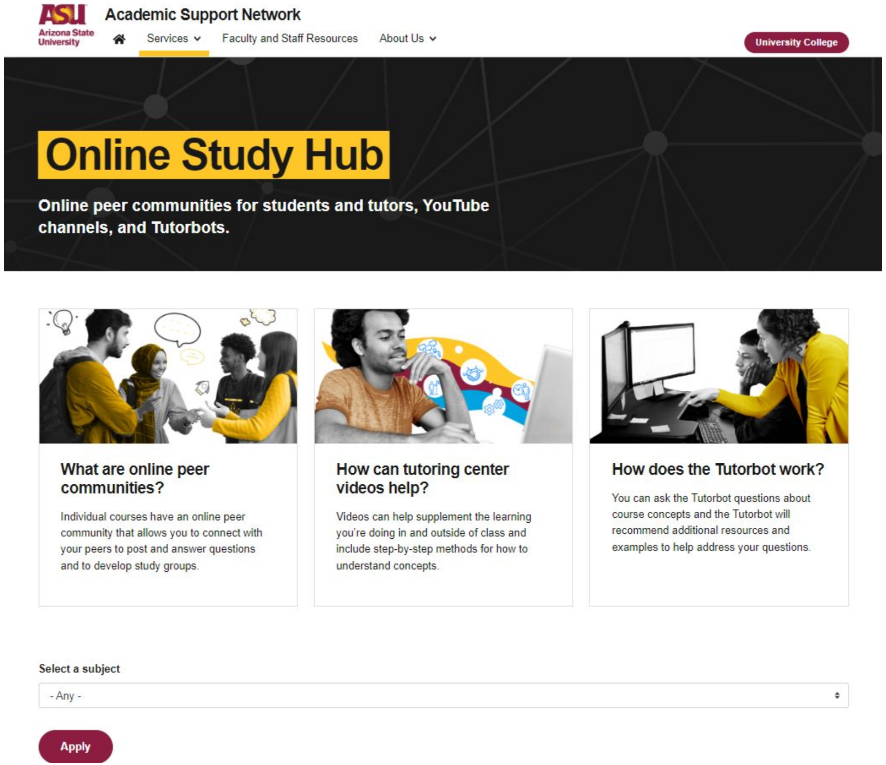
View digital resources

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



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?

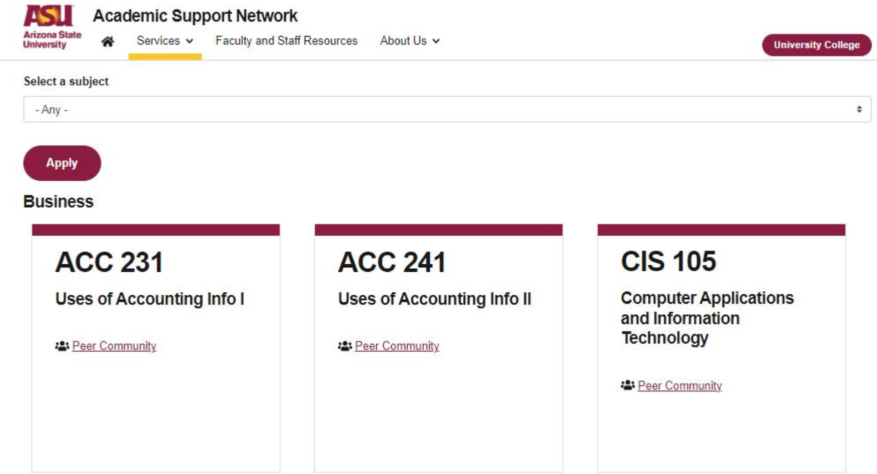
Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.

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



Business

ACC 231 Uses of Accounting Info I — Peer Community

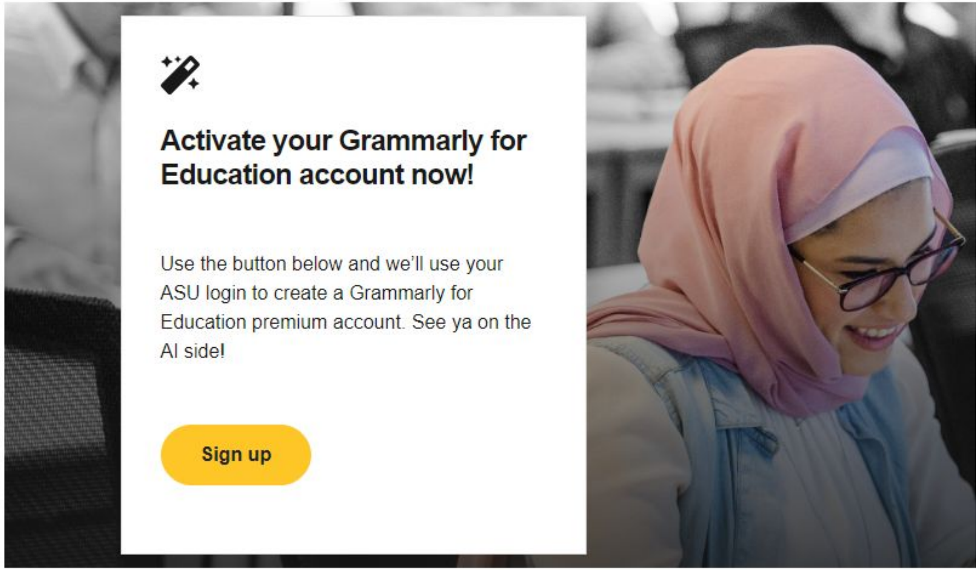
ACC 241 Uses of Accounting Info II — Peer Community

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!

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