SER 321 B Session

SI Session

Sunday March 24th 2024

7:00 pm - 8:00 pm MST

Agenda

- Assignment 3-2 GUI
  - Swing Walkthrough
  - Working with the GUI
- Socket Communication

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

Assign 3-2 Starter Code

🌟 GUI Walkthrough 🌟

Make sure to read the README - it provides a lot of this information, too!

Module 3 was *just* released so if you aren't there yet ***don't stress***!

Shout out to William for his *excellent* comment in #assignment3

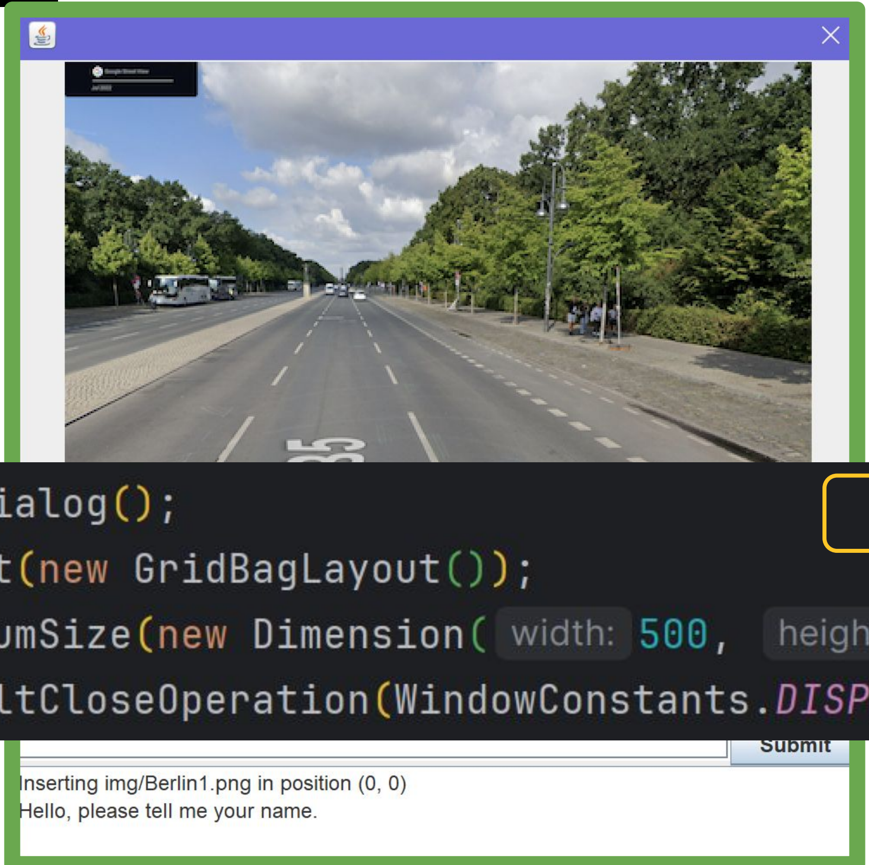
Swing Tutorials

javax.swing package API

SER 321
Assign 3-2 Starter Code

🌟 GUI Walkthrough 🌟

JDialog



```
Inserting img/Berlin1.png in position (0, 0)
Hello, please tell me your name.
```

```java
frame = new JDialog();
frame.setLayout(new GridBagLayout());
frame.setMinimumSize(new Dimension( width: 500,  height: 500));
frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
```

ClientGui.java

[Swing Tutorials](#) [javax.swing package API](#)

SER 321

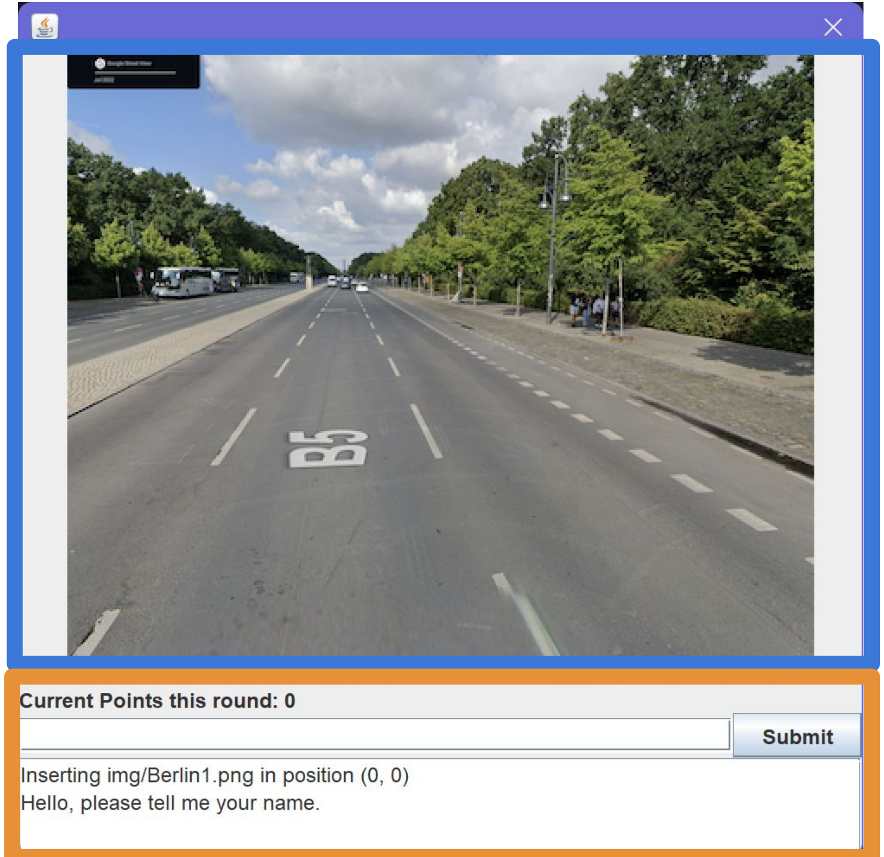
Assign 3-2 Starter Code

🌟 GUI Walkthrough 🌟

JDialog will hold **panels**

picPanel

outputPanel



Current Points this round: 0

```
Inserting img/Berlin1.png in position (0, 0)
Hello, please tell me your name.
```

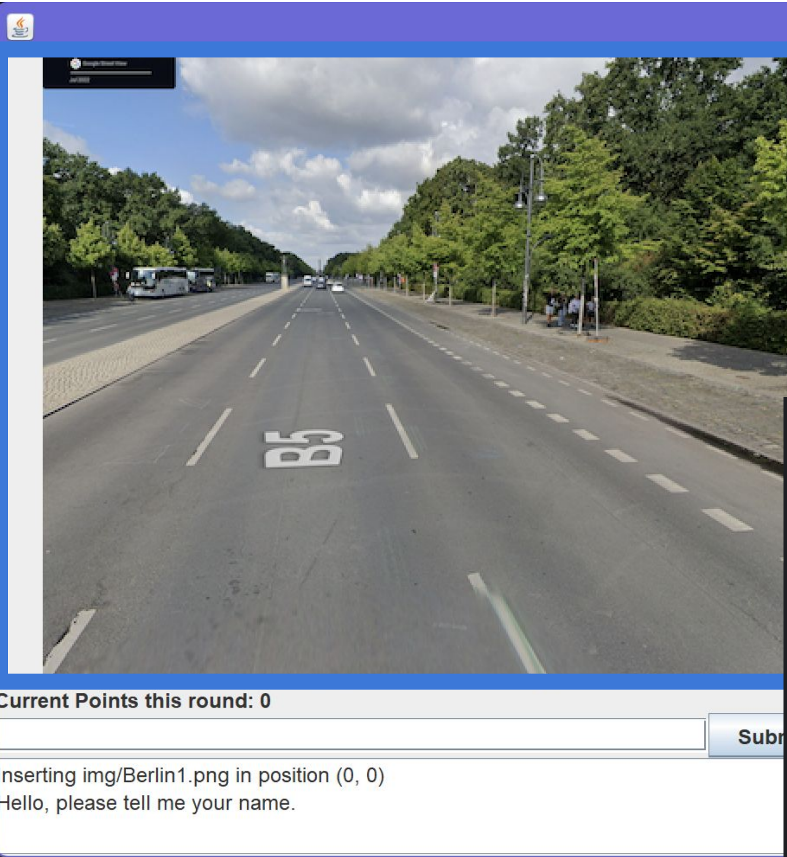
Swing Tutorials

javax.swing package API

SER 321

Assign 3-2 Starter Code

🌟 GUI Walkthrough 🌟

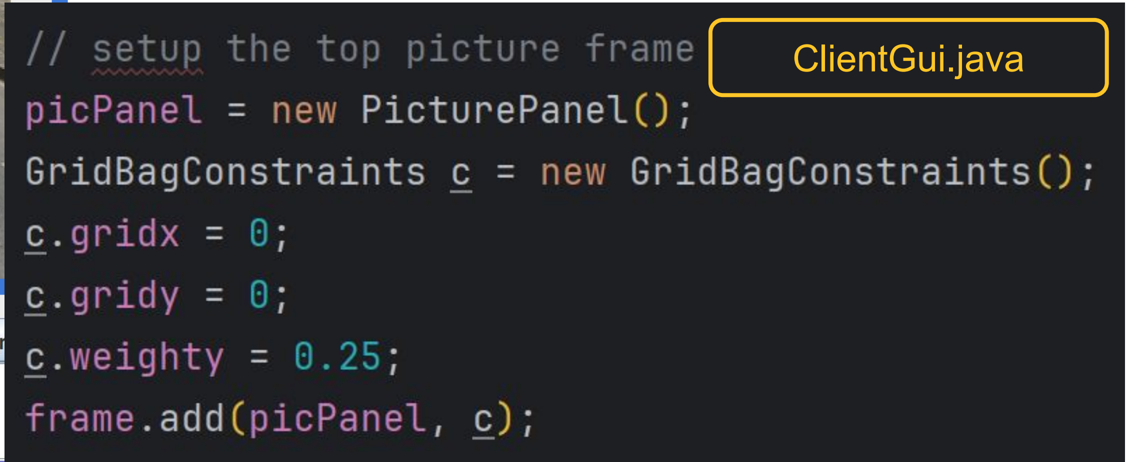


Current Points this round: 0

nserting img/Berlin1.png in position (0, 0)
Hello, please tell me your name.

picPanel

GridBagConstraints is how you define the layout



ClientGui.java

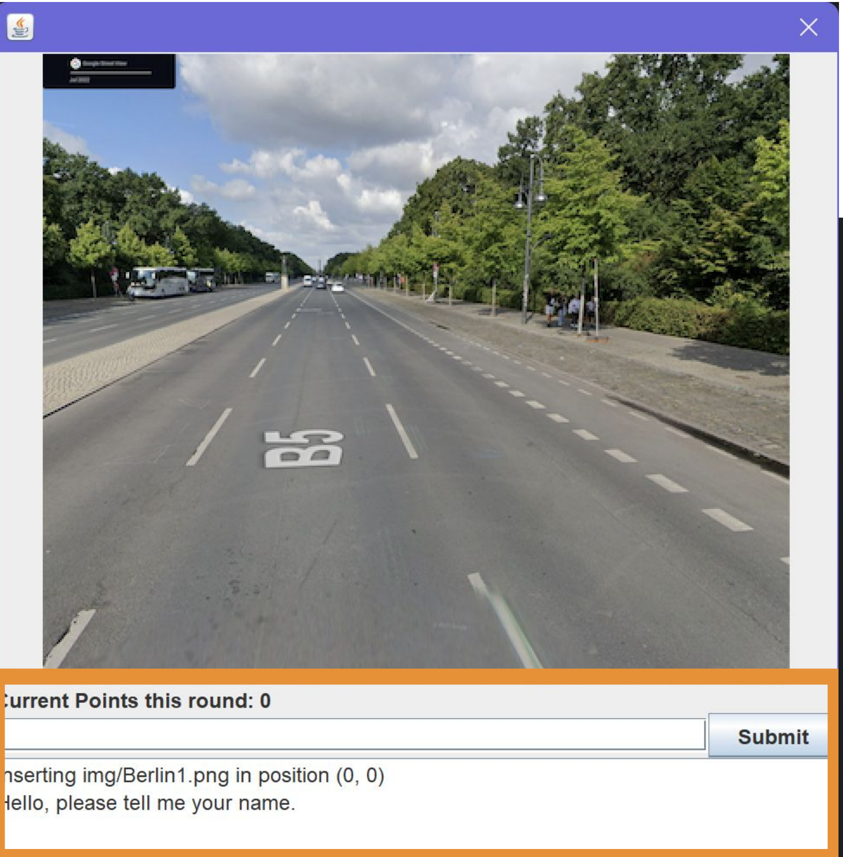
```
// setup the top picture frame
picPanel = new PicturePanel();
GridBagConstraints c = new GridBagConstraints();
c.gridx = 0;
c.gridy = 0;
c.weighty = 0.25;
frame.add(picPanel, c);
```

Swing Tutorials

javax.swing package API

SER 321

Assign 3-2 Starter Code



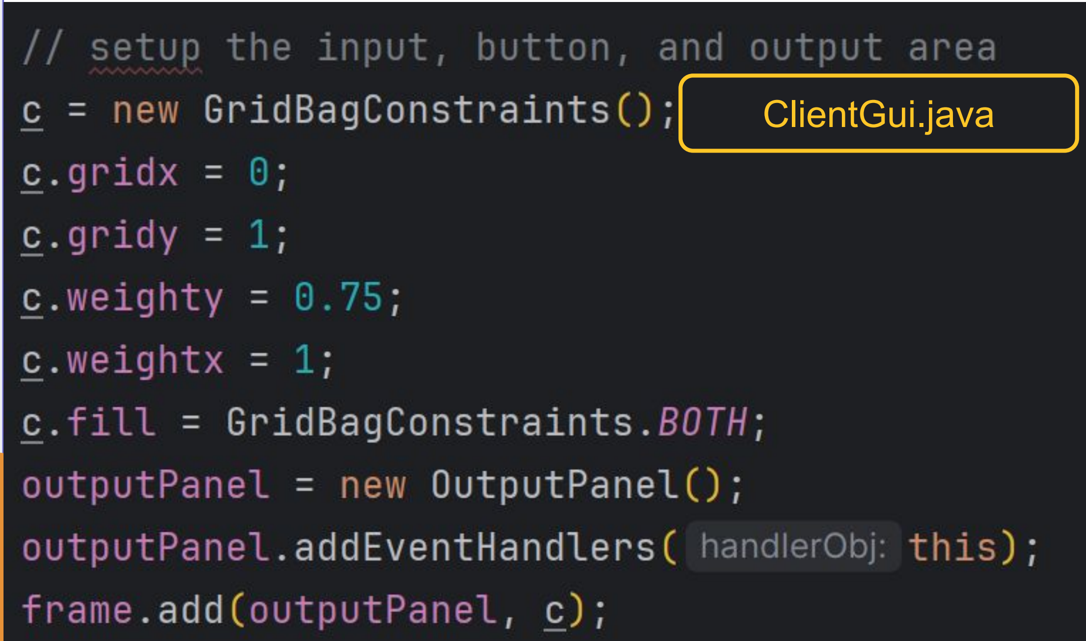
urrent Points this round: 0

nserting img/Berlin1.png in position (0, 0)
ello, please tell me your name.

🌟 GUI Walkthrough 🌟

outputPanel

GridBagConstraints is how you define the layout



```
// setup the input, button, and output area
c = new GridBagConstraints();
c.gridx = 0;
c.gridy = 1;
c.weighty = 0.75;
c.weightx = 1;
c.fill = GridBagConstraints.BOTH;
outputPanel = new OutputPanel();
outputPanel.addEventHandlers( handlerObj: this);
frame.add(outputPanel, c);
```

ClientGui.java

Swing Tutorials

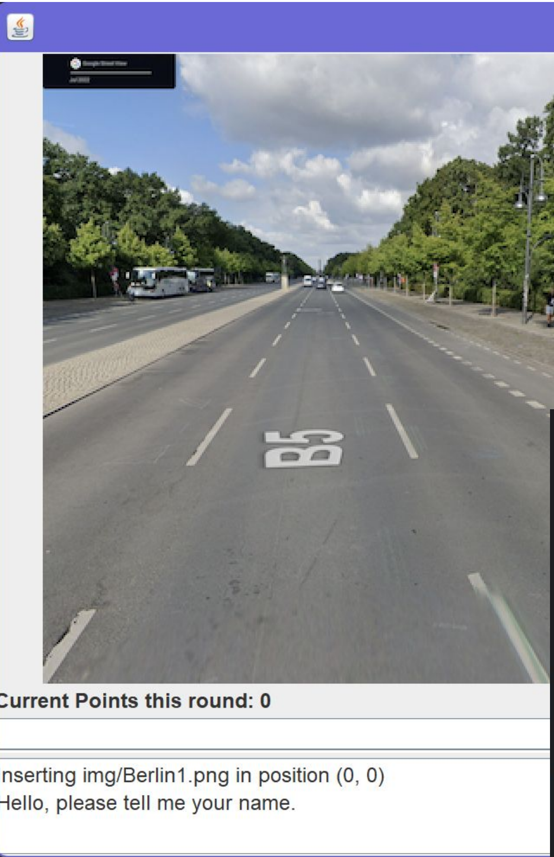
javax.swing package API

SER 321

Assign 3-2 Starter Code

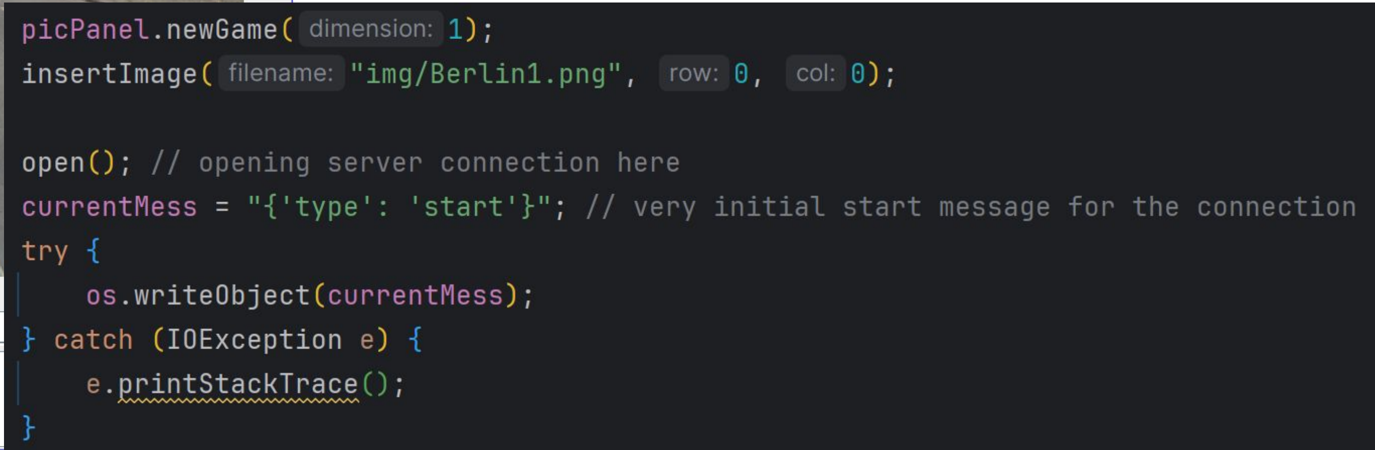
🌟 GUI Walkthrough 🌟

Image is inserted before displaying



Current Points this round: 0

nserting img/Berlin1.png in position (0, 0)
Hello, please tell me your name.



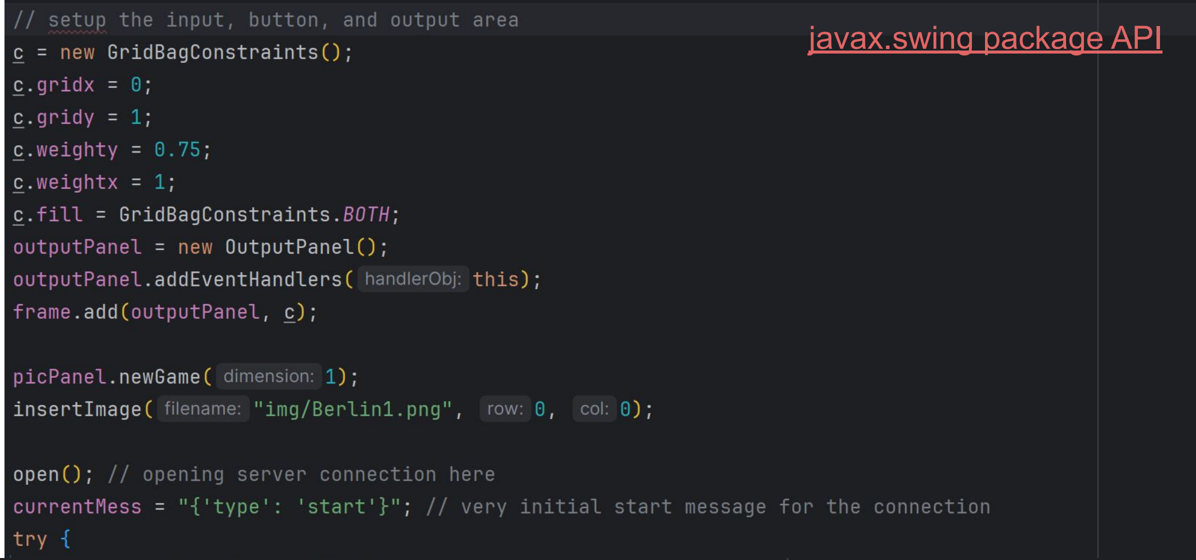
```
picPanel.newGame( dimension: 1);
insertImage( filename: "img/Berlin1.png",  row: 0,  col: 0);

open(); // opening server connection here
currentMess = "{'type': 'start'}"; // very initial start message for the connection
try {
    os.writeObject(currentMess);
} catch (IOException e) {
    e.printStackTrace();
}
```

Swing Tutorials

SER 321

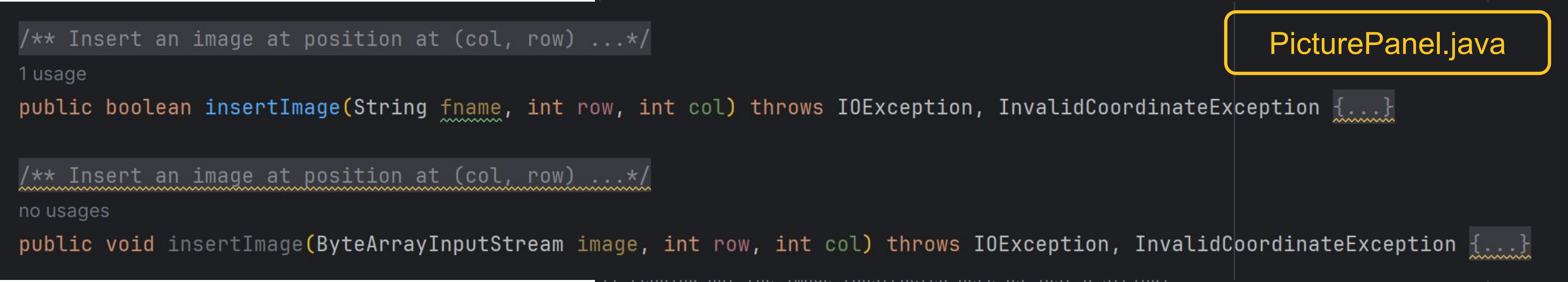
Assign 3-2 Starter Code

insertImage



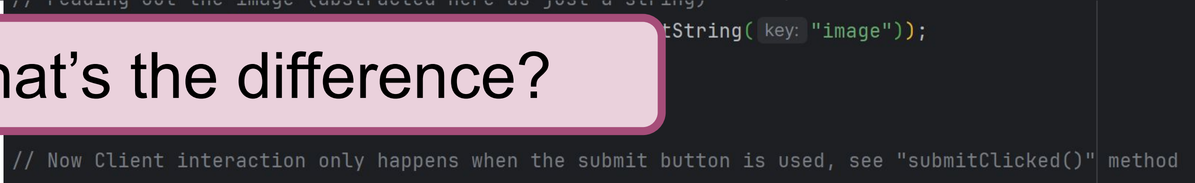
javax.swing package API

```
// setup the input, button, and output area
c = new GridBagConstraints();
c.gridx = 0;
c.gridy = 1;
c.weighty = 0.75;
c.weightx = 1;
c.fill = GridBagConstraints.BOTH;
outputPanel = new OutputPanel();
outputPanel.addEventHandlers( handlerObj: this);
frame.add(outputPanel, c);

picPanel.newGame( dimension: 1);
insertImage( filename: "img/Berlin1.png", row: 0, col: 0);

open(); // opening server connection here
currentMess = "{'type': 'start'}"; // very initial start message for the connection
try {
```



PicturePanel.java

```
/** Insert an image at position at (col, row) ...*/
1 usage
public boolean insertImage(String fname, int row, int col) throws IOException, InvalidCoordinateException {...}

/** Insert an image at position at (col, row) ...*/
no usages
public void insertImage(ByteArrayInputStream image, int row, int col) throws IOException, InvalidCoordinateException {...}
```



```
// reading out the image (abstracted here as just a string)
tString( key: "image"));

// Now Client interaction only happens when the submit button is used, see "submitClicked()" method
```

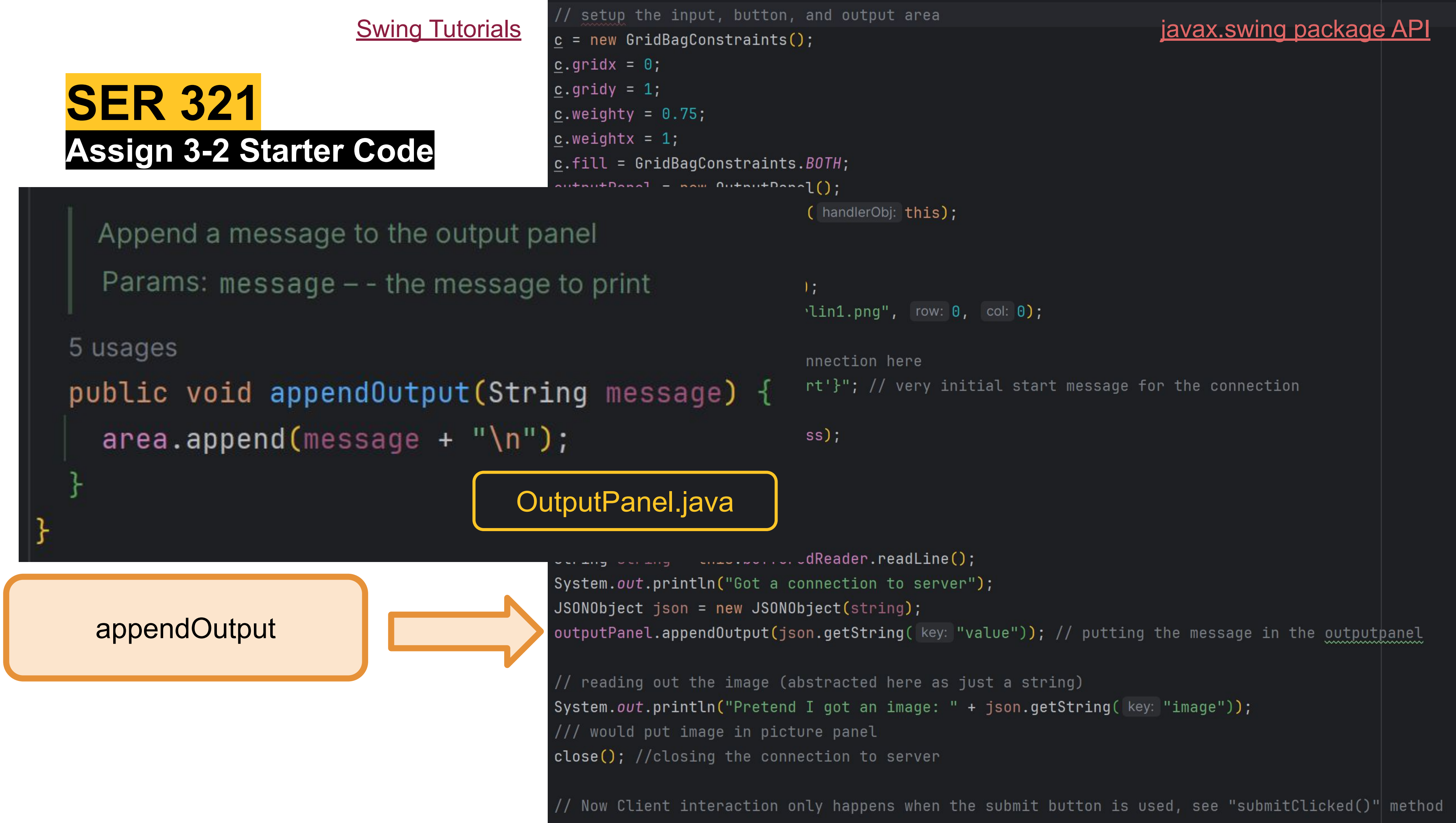
What’s the difference?

Swing Tutorials

SER 321

Assign 3-2 Starter Code

javax.swing package API



```
Append a message to the output panel
Params: message – - the message to print
5 usages
public void appendOutput(String message) {
  area.append(message + "\n");
}
}
```

OutputPanel.java

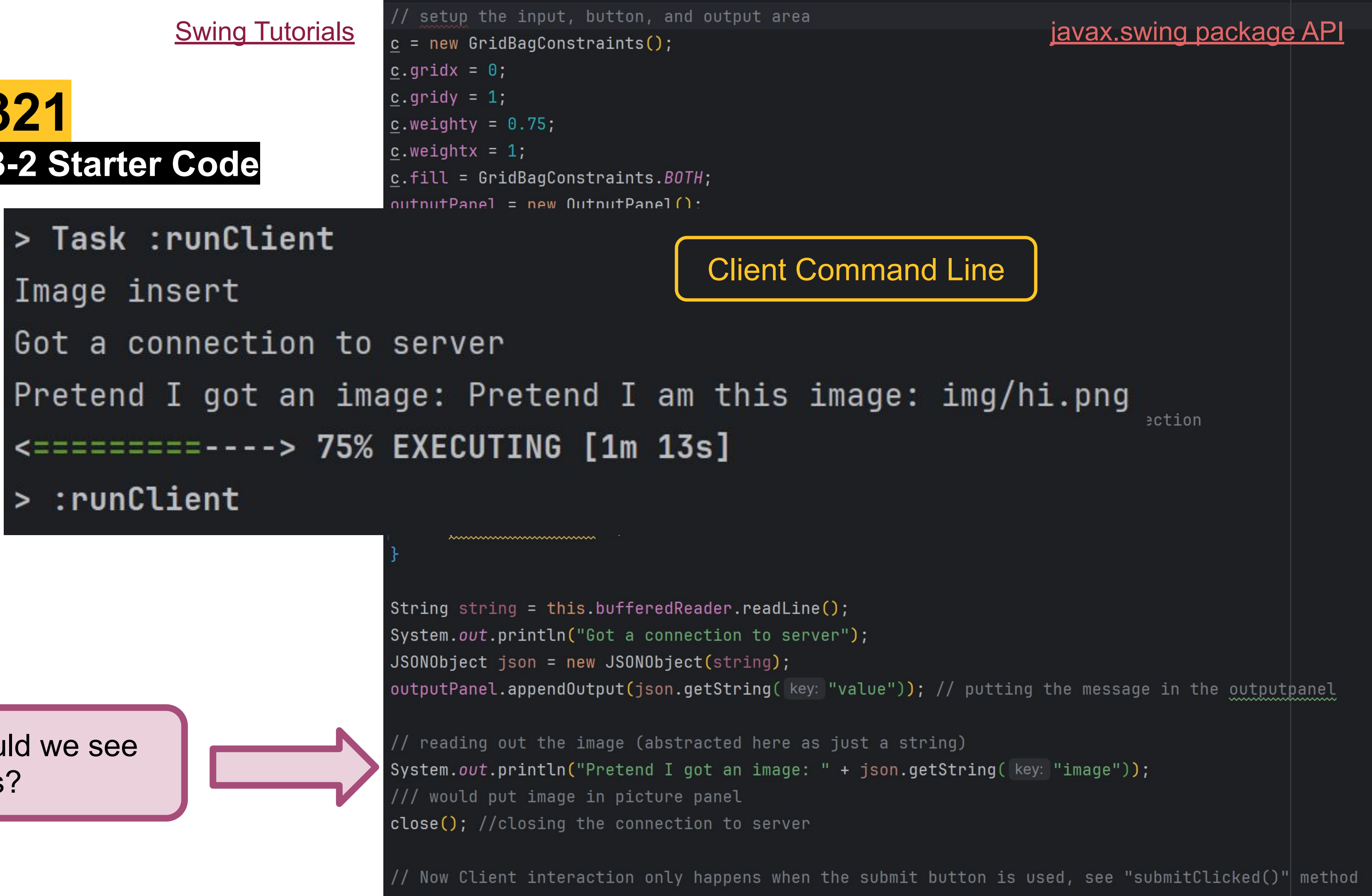
```
// setup the input, button, and output area
c = new GridBagConstraints();
c.gridx = 0;
c.gridy = 1;
c.weighty = 0.75;
c.weightx = 1;
c.fill = GridBagConstraints.BOTH;
outputPanel = new OutputPanel();
( handlerObj: this);

);
lin1.png", row: 0, col: 0);

nnection here
rt'}"; // very initial start message for the connection

ss);

dReader.readLine();
System.out.println("Got a connection to server");
JSONObject json = new JSONObject(string);
outputPanel.appendOutput(json.getString( key: "value")); // putting the message in the outputpanel

// reading out the image (abstracted here as just a string)
System.out.println("Pretend I got an image: " + json.getString( key: "image"));
/// would put image in picture panel
close(); //closing the connection to server

// Now Client interaction only happens when the submit button is used, see "submitClicked()" method
```

appendOutput

[Swing Tutorials](#)

SER 321

Assign 3-2 Starter Code



```java
// setup the input, button, and output area
c = new GridBagConstraints();
c.gridx = 0;
c.gridy = 1;
c.weighty = 0.75;
c.weightx = 1;
c.fill = GridBagConstraints.BOTH;
outputPanel = new OutputPanel();
```

```
> Task :runClient
Image insert
Got a connection to server
Pretend I got an image: Pretend I am this image: img/hi.png
<=========----> 75% EXECUTING [1m 13s]
> :runClient
```

```java
ection
}

String string = this.bufferedReader.readLine();
System.out.println("Got a connection to server");
JSONObject json = new JSONObject(string);
outputPanel.appendOutput(json.getString( key: "value")); // putting the message in the outputpanel

// reading out the image (abstracted here as just a string)
System.out.println("Pretend I got an image: " + json.getString( key: "image"));
/// would put image in picture panel
close(); //closing the connection to server

// Now Client interaction only happens when the submit button is used, see "submitClicked()" method
```

[javax.swing package API](#)

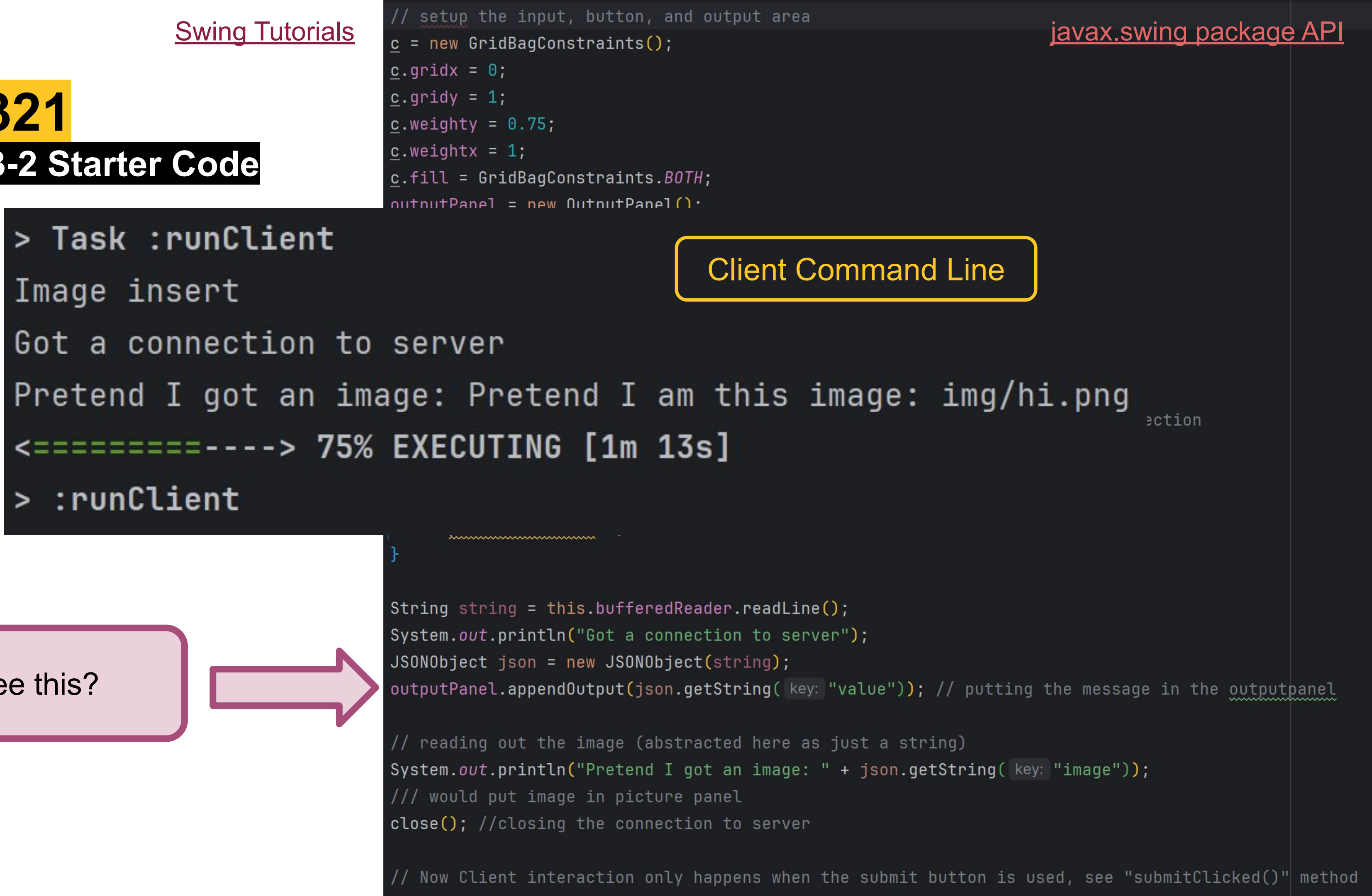
Client Command Line

Where should we see this?

Swing Tutorials

SER 321

Assign 3-2 Starter Code



```
// setup the input, button, and output area
c = new GridBagConstraints();
c.gridx = 0;
c.gridy = 1;
c.weighty = 0.75;
c.weightx = 1;
c.fill = GridBagConstraints.BOTH;
outputPanel = new OutputPanel();
```

```
> Task :runClient
Image insert
Got a connection to server
Pretend I got an image: Pretend I am this image: img/hi.png
<=========----> 75% EXECUTING [1m 13s]
> :runClient
```

```
}

String string = this.bufferedReader.readLine();
System.out.println("Got a connection to server");
JSONObject json = new JSONObject(string);
outputPanel.appendOutput(json.getString( key: "value")); // putting the message in the outputpanel

// reading out the image (abstracted here as just a string)
System.out.println("Pretend I got an image: " + json.getString( key: "image"));
/// would put image in picture panel
close(); //closing the connection to server

// Now Client interaction only happens when the submit button is used, see "submitClicked()" method
```

javax.swing package API

Client Command Line

Do we see this?

Swing Tutorials

javax.swing package API

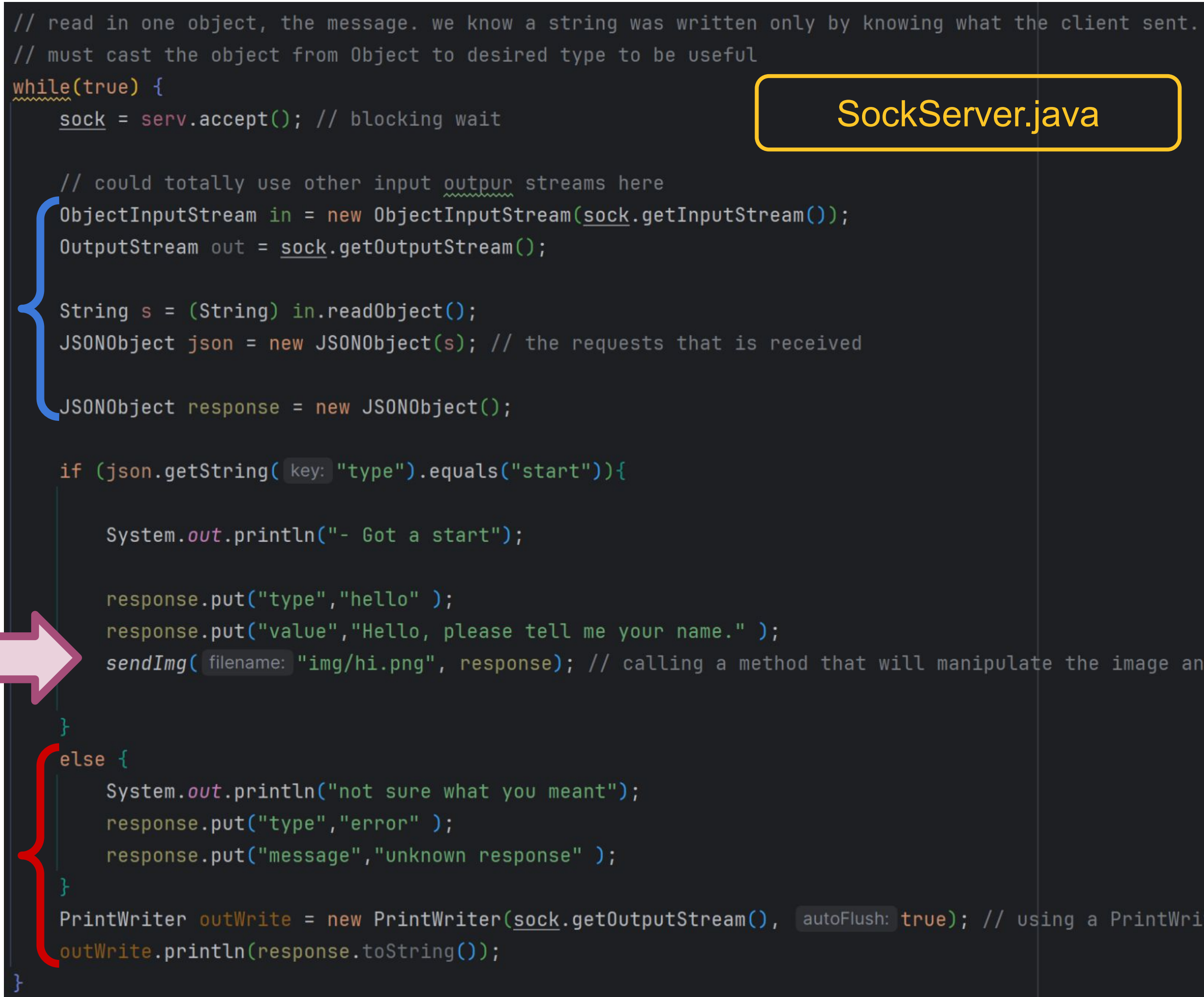
SER 321

Assign 3-2 Starter Code

Read Client Request

Send image to Client

Basic Error Catch



SockServer.java

```
// 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
while(true) {
    sock = serv.accept(); // blocking wait

    // could totally use other input outpur streams here
    ObjectInputStream in = new ObjectInputStream(sock.getInputStream());
    OutputStream out = sock.getOutputStream();

    String s = (String) in.readObject();
    JSONObject json = new JSONObject(s); // the requests that is received

    JSONObject response = new JSONObject();

    if (json.getString( key: "type").equals("start")){

        System.out.println("- Got a start");

        response.put("type","hello" );
        response.put("value","Hello, please tell me your name." );
        sendImg( filename: "img/hi.png", response); // calling a method that will manipulate the image an

    }
    else {
        System.out.println("not sure what you meant");
        response.put("type","error" );
        response.put("message","unknown response" );
    }
    PrintWriter outWrite = new PrintWriter(sock.getOutputStream(),  autoFlush: true); // using a PrintWri
    outWrite.println(response.toString());
}
```

SER 321

Assign 3-2 Starter Code

Let’s take a look at the starter code together for full context

SER 321

Scratch Space

Questions?

Survey:

http://bit.ly/ASN2324

Upcoming Events

SI Sessions:

- Monday, March 25th at 7:00 pm MST
- Thursday, March 28th at 7:00 pm MST
- Sunday, March 31st at 7:00 pm MST

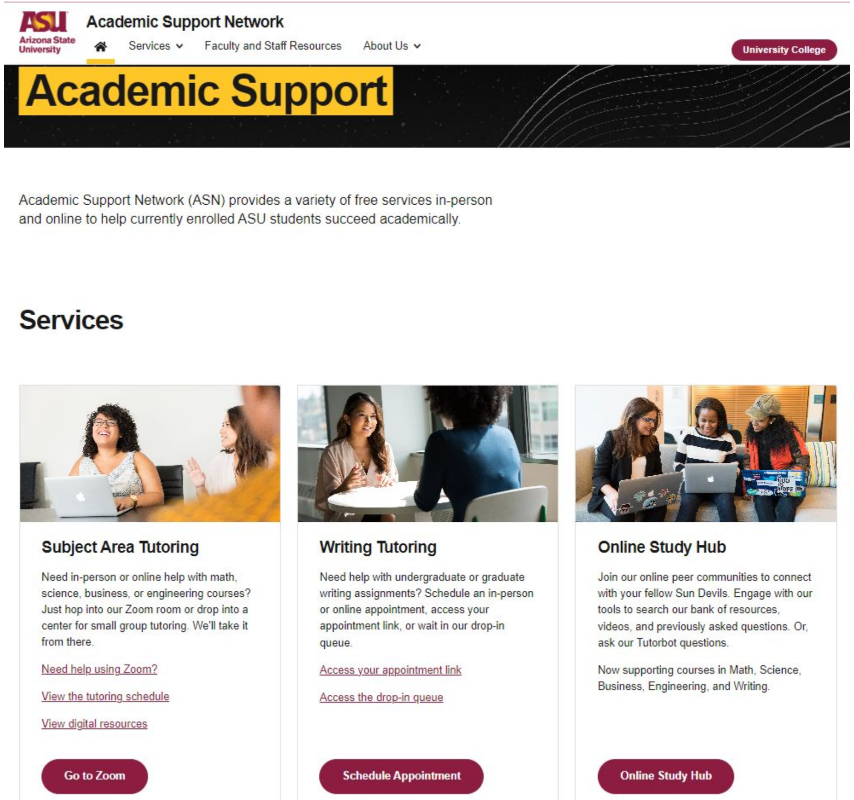
Review Sessions:

- 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

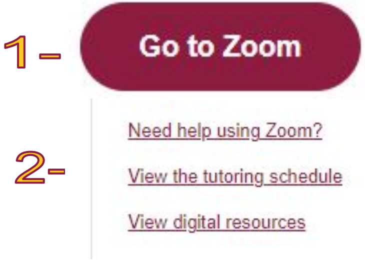
Schedule Appointment

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

2- Need help using Zoom?

View the tutoring schedule

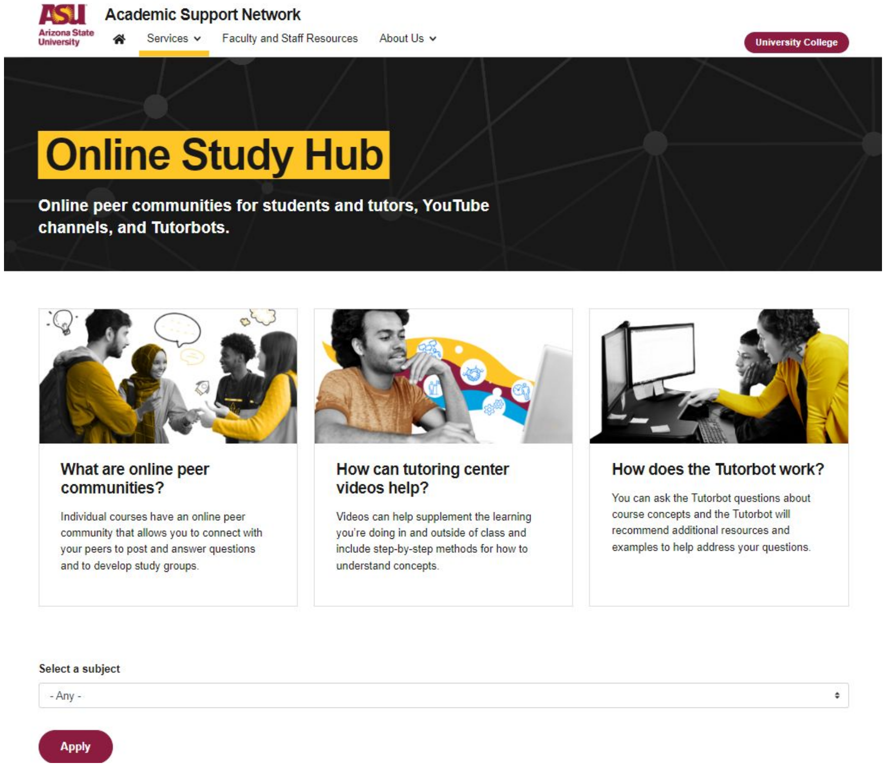
View digital resources

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



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.

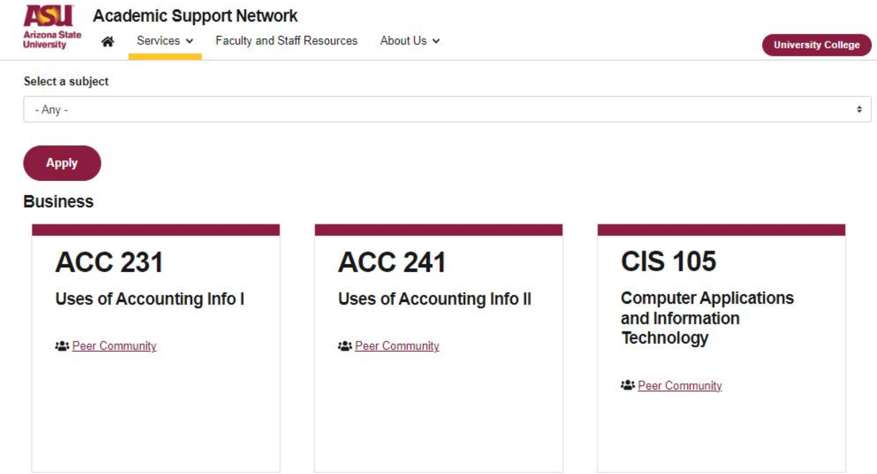
How can tutoring center videos help?

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 -



Select a subject: - Any -

Business

ACC 231 Uses of Accounting Info I — Peer Community

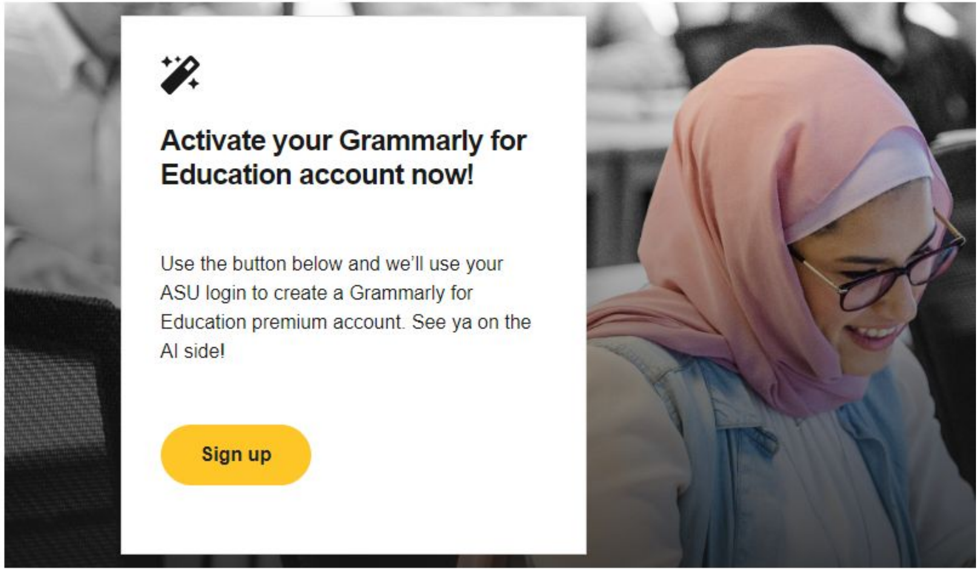
ACC 241 Uses of Accounting Info II — Peer Community

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
- javax.swing package API
- Swing Tutorials