Class 5: Using Processing with JavaMail API Part 2

We will look at some code and write some code.

Using Gmail API

- Go over the API
- Look at your DoSomething() functions
- Send email back
- If time permits, add Arduino

Main structure of an email:

- Sender (constant)
- Subject (variable)
- Body (variable)

Assignment due Wednesday

Integrate Email processing code with one of your Arduino projects

```
void sendMail(String to, String from, String subject, String body)
{
  Properties props=new Properties();

  // SMTP Session for Gmail
  props.put("mail.transport.protocol", "smtp");
  props.put("mail.smtp.host", smtp_host);
  props.put("mail.smtp.port", "587");
  props.put("mail.smtp.auth", "true");

  // TLS setting for Gmail
  props.put("mail.smtp.starttls.enable","true");

  // Creates new session
  Session send_session = Session.getInstance(props, null);

  try
  {
    // Create new message
    MimeMessage message = new MimeMessage(send_session);

    // Define sender
    message.setFrom(new InternetAddress(email, from));

    // Define recipient
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to,
false));

    // Set the subject
    message.setSubject(subject);

    // Set body
    message.setText(body);

    //Authenticate to the SMTP server
    Transport transport = send_session.getTransport("smtp");
    transport.connect(smtp_host, 587, email, pass );
```

```
    //Send E-mail
    transport.sendMessage(message, message.getAllRecipients());

    //Close connection
    transport.close();
    println("Mail sent to: " + to);
}
  //Basic error handler
  catch(Exception e)
  {
    e.printStackTrace();
  }
}
```