Interactive Extraction of Examples from Existing Code

Auxiliary Material

A1. Observations from Formative Study

We conducted a formative study to understand the process that programmers follow when creating executable code examples from their own code, and the obstacles they encounter along the way. We observed 12 programmers as they created example code (referred to as F1–12 for “formative study participant”). Participants were recruited from our professional networks, local MeetUps, and computer science researchers from our institution.

Participants were asked to create an understandable, executable code example from code they had recently written. The first five participants were asked to produce this code example as a blog post. We observed that the first five participants sometimes produced erroneous or untested code. With this observation, we realized that, without the proper tools, examples might be created that contain errors and do not run. We asked all remaining participants to only create a minimal, understandable, executable code example.

Participants authored code examples in a variety of languages, including PHP, Java, Python, JavaScript, Bash, C, and C#. Code examples demonstrated a variety of tasks, including querying a Salesforce database, and building a Chrome extension for circumventing news website pay walls. One participant (F12) unexpectedly brought code that had been written by someone else that they wished to simplify; because they followed a similar workflow as other participants, we include anecdotes from their experience as well.

Theme 1: Supporting Additive Authoring of Code Examples

When we cued the first five participants to produce a blog post that demonstrated a usage pattern with concise, easy-to-understand example code, all participants took an “additive” process to authoring example code, copying and pasting code from an existing project into a new
Programmers copied and pasted individual lines (F2, F4, F6) and the contents of entire files (F1, F3) into a new buffer.

Participants who built a code example by moving code from an existing project faced two major obstacles. First, in some cases, it could be tedious to rebuild the original code's project settings, dependencies, and environment for the code example. For F6, this involved generating configuration files, setting paths to libraries, copying import statements, duplicating style files, and modifying the directory structure. All of these steps had to be completed before any example code could be tested. The same code could encounter unexpected bugs in a new location. For example, relative file paths resolved to the wrong location when the code was run in a different directory (F11).

Second, programmers introduced bugs in the process of copying, pasting, and modifying the code. Sometimes, important fragments of code were missing that caused the program to crash, like missing variable definitions or return statements (F2, F4). Programmers introduced bugs through manual edits that, while appearing innocuous, altered the program's behavior in unexpected ways (F4, F10).

After the first participants faced obstacles when authoring example code additively, we asked F8–12 to author code examples “subtractively”, by removing code from a working project until they created a minimal code example. While participants no longer needed to configure the runtime environment for a new example, removing unrelated code and dependencies could still take tens of minutes (F10, F12). This confirmed for us that, for both approaches, the demonstrated authoring techniques were costly and tedious.

**Theme 2: Modifying Examples to Improve Effectiveness**

Participants omitted irrelevant implementation details to better highlight the focal code of an example. To reduce irrelevant detail, participants simplified strings to “Lorum Ipsum” (F5) and “hello world” strings (F7), and simplified the SQL queries to include only fields used by the final example code (F8). Return types of methods were changed from complex types to simpler types like lists (F8). Literal values were inserted to provide concise and reasonable default parameters (F8, F12). Participants commented out the names of argument parameters that would distract from those that the user should set, while leaving them present in the code.
to prevent compiler errors (F4), and annotated suggested values for arguments (F2, F4).

We observed that authors sometimes added code back in to make the example more usable. By adding code, authors could increase the visibility of execution, demonstrate best practices, and provide variations on examples. To make the code example’s internal computations more visible, participant F7 replaced a return statement with a print statement to make the code’s success obvious to someone who executed it, and F12 left in dozens of lines dedicated to printing out updates to a machine learning model during training. F11 added explicit if statements to check whether method return values were null pointers and empty lists, to make clear to readers that the API calls might return empty values. F4 produced three versions of one code example, each of which utilized different test data and showed progressively more advanced usage of an audio API.

A2. System Implementation Details

As described in the “System Implementation” section of the main text, CodeScoop comprises a collection of modules to flag errors, suggest resolutions, and resolve errors. This section describes additional modules that were not described in the main text.

Detecting errors and relevant code

Missing previous uses of a variable

When a user adds a variable use and its definition, CodeScoop collects all past uses of the variable that occur between the use and definition, and which are not yet in scoop. These previous uses are presented to the user all at one time.

Missing exceptions for a function call

Using Soot as a parser, we locate all function calls in the source program. For each call, we use the Java Reflections API¹ to find all exceptions the method can throw. An ANTLR parser is run on the source program to find try-catch blocks and the exceptions they catch. When a

¹https://docs.oracle.com/javase/tutorial/reflect/index.html
line is added to the scoop with an exception-prone function, CodeScoop checks to see whether (a) the scoop already throws the exception or a superclass of the exception or (b) the scoop includes a try-catch block surrounding the function call that catches that exception. If neither of these is true, CodeScoop reports that this exception must be caught or thrown.

**Suggesting fixes and code additions**

*Throwing exceptions for exception-prone function calls*

CodeScoop searches for an import statement that defines either the exception or one of its superclasses. When a relevant type is found in the import statements, CodeScoop recommends that this exception type is thrown from the main method. If an user accepts the suggestion, the exception type is added to the throws clause of the main method.

*Suggesting stubs for undefined object variables*

CodeScoop extends the idea of replacing variable names with literal values to complex data types, by generating “stubs” with identical runtime behavior to object variables. While this feature is currently experimental, we propose it as one possible solution. To create stubs, an additional debugger virtual machine is launched. CodeScoop tracks every property access and every method call on every object defined in the source program, and logs all values these properties and methods returned. These variables and method calls can return other objects, which get tracked as well. CodeScoop generates stubs as classes that replicate the recorded behavior given the same order of property accesses and method calls.

**Rendering Example Text from an Internal Scoop**

Generating the example text from a list of selections and user choices involves instantiating a hierarchical string template. First, the text for all text selections, stored as numerical “ranges” that indicate the character offset of the text selections in the source program, is retrieved from the source program. These texts are joined in a temporary buffer, in the order that they appeared in the original code. The position of each of these selections is “marked” in the
example text editor. These markers will keep track of the selection’s offset in the example text as it is built up, and will be used later for highlighting errors and inserting literal values.

Next, the snippets are wrapped in the `main` function of a class called `ExtractedExample`. Object stubs are generated from inner classes and added above the `main` function. All inner classes and local methods that the user included are added as static members of the class below the `main` function. Finally, the generated example code is automatically indented.

Once all selections and stubs have been added, literal values are inserted into the example, in the positions tracked by the “markers” of individual text selections. If the program is in the flag state, it then highlights errors for a user to choose from. If in the suggest state, it highlights the chosen error and displays a menu for previewing and accepting resolutions.

### A3. CodeScoop Usability Study Tasks

For each example extraction task, participants were shown one of three fabricated “Stack Overflow questions”, describing something a programmer might want to do with Java. Participants were asked to make an example that answered the question they were shown. Each question was shown as a mocked-up post on the Stack Overflow site, which included a title, body, author, and vote count. For example, this post was shown as a prompt for task 1:

![Stack Overflow Mockup](image)

The titles and texts of the posts for the three tasks were as follows:

- **Task 1.** How do you get a row from a database in Java? I’m currently trying to use the...
Cursor class from the org.acme.database package. I can't find any examples about it. What's the recommended way of get a row from a table in the database?

Task 2. **Using JSoup to extract text from elements on a web page?** I'm writing a program to fetch a web page and want to extract the text from all of a certain type of element. I know I can use JSoup for this, but I'm stuck on how exactly to do it.

Task 3. **Sending an email over SMTP with Java.** Does anyone have a suggestion for how to send an email message automatically in Java? I'm trying to use the 'javax.mail' package, but haven't figured out how to get it to work yet.

Because participants had a limited time to review code before extracting an example, we offered short written hints of what they should look for in the source programs, and what should be included in the examples they extracted. We introduced these checklists to participants as “bare minimum checklists”, to emphasize participants that participants should include more than this minimum if they felt it was appropriate. The texts of the checklists for each task were:

- **Task 1.** You will probably need to make a Database, get a cursor for the database, fetch a line, and get data from that line. Beyond that, use your best judgment.
- **Task 2.** You will probably need to create a JSoup object for a URL and extract links from a JSoup object. Beyond that, use your best judgment.
- **Task 3.** You will probably need to create a session, make a message with some text and a recipient, connect to a transport and send a message. Beyond that, use your best judgment.

Participants were also provided with a set of guidelines of verbal cues they should use to indicate when they were finished scanning the source program or extracting the example. The written guidelines, reviewed verbally with each participant, were:

**Instructions.** For each question you are given,

- Take 5 minutes to find and understand relevant parts of the code. When you're done, say “I understand the relevant parts of the code.”
Participants were allowed to ask the study facilitator any questions they wished about how the source programs worked at any time, with the intent of helping participants focus on extracting the right example rather than comprehending unfamiliar code.

A4. CodeScoop Tutorial Task

Before performing any of the main experimental tasks, participants were asked to complete a tutorial that taught them how to use the major features of CodeScoop (nicknamed “Rockies” in the study). We include the guide for the tutorial as a document called “How to Use Rockies”, starting on the next page.

A5. Source Programs and Scooped Examples

In the usability study, participants were asked to create examples from two source programs. We list the source programs and the 19 examples that participants created with CodeScoop beneath the CodeScoop tutorial guide.
How to Use Rockies

Rockies is a tool for making example code from existing code.

Suppose you’ve been working on the code in `SillyDictionaryBuilder.java` (see below). The program generates random words and saves them to a file.

Your friend asked you how to create a random string of alphabetic characters. You want to share your example with your friend. But you also want to make it shorter and more readable, while keeping it compilable and runnable.

In this tutorial, you will use Rockies to extract an example that is more concise than the original code, but also runnable. If you learn two things about Rockies in this tutorial, it is that:

1. Rockies helps you include necessary code to preserve program behavior
2. Rockies proposes values from the original program’s execution to give you opportunities to eliminate irrelevant or complex code

Start an Example by Pointing to Main Example Results

First, select the code that should have the final results of the example program. In this case, it’s the text on Line 28, which wraps up the creation of a random word. Select this line:
To extract a code example, right click on the selected text. Scroll down and click on the “Make example” command:

Immediately, you’ll notice that Rockies creates a new pane for the extracted example:
Rockies Makes Some Fixes For You

Rockies takes about 10-20 seconds to analyze the code after you’ve made the initial selection. Once it’s done analyzing the code, it (1) starts fixing the code and (2) starts suggesting ways for you to complete the example.

For example, you’ll see that one line was already added that defines the variable “buffer”. There’s also a suggestion of more updates to make to the code. Ignore this suggestion for now: we’ll get to it in a few steps.
Compile and Run the Code

Before interacting with Rockies, let's check on the state of the example code in the right pane: You can compile and run the code at any time by clicking on the Run button. Do it right now to see whether the compiles and runs:

Once the code has finished running, you'll see any output in the pane at the bottom. The example doesn't yet print any results, so all you see is a “Finished” message:
Adding Other Uses of a Variable

Rockies prompts you to consider other lines that use variables you’ve included, to make sure you don’t leave out any important logic. This is what it’s doing with the ‘Do you want any of those uses of “buffer”?‘ prompt.

Rockies highlights the lines on the left side that use the variable. Click on the line number in the left text editor to include that line, by clicking on the number 26:

For future prompts about variable uses, if you don’t want to include any of the suggested lines, just click “No” to the right of the prompt.

Tip: You can include multiple lines at once by clicking and dragging over the line numbers.
Adding Missing Code

Rockies highlights undefined variables. You'll need to add in extra code or data to resolve undefined variables before the example can compile and run. Currently, character is undefined. See it highlighted?

To define a variable, move the mouse underneath that variable. See the “Define” button appear:
This pulls up a small menu that will let you define the variable by either adding code or setting a concrete value for the variable. Hover over “Line 25” in the “Add code” menu. This highlights the line in SillyDictionaryBuilder.java that you should include to define character.

```
public class SillyDictionaryBuilder {
    public static void main(String[] args) throws IOException {
        int MIN_ASCII_CHARACTER = 65;
        int ALPHABET_LENGTH = 26;
        int WORD_LENGTH = 8;
        int NUM_WORDS = 30;
        List words = new ArrayList();
        Random randomGenerator = new Random();
        for (int i = 0; i < NUM_WORDS; i++) {
            StringBuffer buffer = new StringBuffer();
            for (int c = i; c < WORD_LENGTH; c++) {
                int asciiCode = MIN_ASCII_CHARACTER + randomGenerator.nextInt(26);
                buffer.append((char) asciiCode);
            }
            String word = new String(buffer);
            words.add(word);
            System.out.println("Saving the dictionary to file..."+word);
            PrintWriter writer = new PrintWriter("outputList.txt");
            writer.println(word);
        }
    }
}
```

Click on “Line 25”, and the line will be automatically added.

Repeat this to define the variable asciiCode, by adding Line 24.
Reviewing Structure Suggestions

When you choose code both inside and outside of a control structure (if statements, try-catch blocks, for loops, etc.), Rockies asks if you want to include that control structure:

- **Accept** the first suggested “for” loop: this one lets you build a word from characters
- **Reject** the second suggested “for” loop: it creates multiple words, which isn’t necessary to the example. We only want to build up one word
Replacing Variables with Values

Sometimes, you don’t want to add code to resolve an undefined variable: It might look better to just add a literal value in the place of that variable.

This time, when you define, `MIN_ASCII_CHARACTER`, hover over the “Set value” menu:

Hover the mouse over the value “65”. You can see the value directly substituted into the example program for you to preview it:
Click on “65” to replace the variable with the value.

Repeat this process to:

- Replace `ALPHABET_LENGTH` with 26
- Replace `WORD_LENGTH` with 8

Print Out a Variable

While this example code can compile and run, it doesn’t show anything. Rockies can help you add a print statement for variables to display output at the end of an example.

In the example pane, select the text for the variable `word` in the line: “String word = new String(buffer);”, and then click the “Print” button.

Why is there a button for printing? A print button lets you display output, without the need for typing or pasting in hand-written text. Rockies only allows expansion of the code through line selections, prompts, and these print statements, to make sure it can still recommend a path to completing the example no matter what state it is in.
You'll see that a print statement was added for the variable:

```java
import java.util.Random;

public class ExtractedExample {

    public static void main(String[] args) {
        Random randomGenerator = new Random();
        StringBuffer buffer = new StringBuffer();
        for (int c = 0; c < 8; c++) {
            int asciiCode = 65 + randomGenerator.nextInt(26);
            char character = (char) asciiCode;
            buffer.append(Character.toString(character));
        }
        String word = new String(buffer);
        System.out.println(word);
    }
}
```

Run the example again to verify that the output is now shown.
Throwing Relevant Exceptions

While the example is now complete, there's one more feature you should know about. Rockies can help you “throw” relevant exceptions when for lines of code you included. Let's try this out. Click on Lines 33 and 39 in the left pane:

Line 33 can throw an IOException. Rockies tells you this by pointing to the line as it appears in the example editor:
Click on “Accept” to throw the exception. If you don’t do this, and the IOException is not caught by a try-catch block, the code may not compile.

Undo the Last Action

With Rockies, the only way to remove lines of code is to rewind the example-making process. To do this, click the *Undo* button. Click it once just to try it.

---

Parting Words

If you learn two things about Rockies in this tutorial, it is that:

1. Rockies **helps you include necessary code to preserve program behavior**
2. Rockies proposes values **from the original program's execution** to give you opportunities to **eliminate irrelevant or complex code**

To have maximum success with Rockies, keep in mind that **Rockies helps the most when your initial code are the lines including the final results of the example**. This helps Rockies identify other lines of code you may want to include.
For each example that a participant produced with CodeScoop, we display the example with annotations of how it was constructed.

The marks in the left gutter indicate author interactions. There are three types of interactions:

- **Manual**: The participant included these lines without any prompting.
- **Boilerplate**: This code shows up in every example so the example can compile.
- **Prompted**: The participant included these lines when reviewing a prompt from CodeScoop.

The labels in the right gutter are fine-grained descriptions each line’s source. There are three variants of prompted selections:

- **def**: adding code to define a variable
- **use**: adding a previous use of a variable
- **control**: adding a control structure that surrounds a statement
- **throw**: throwing an exception from the main method signature

A line is generated with the "add print statement" label when an author clicks the "Print" with a variable name selected.

An author replaced an undefined variable with a literal, the literal is highlighted in green, underlined, and bolded.
import org.acme.database.Book;
import org.acme.database.Booklist;
import org.acme.database.Cursor;
import org.acme.database.Database;
import java.util.ArrayList;
import java.util.List;

public class BookListing {
    private boolean DEBUG = true;

    public Booklist getBookListing(String genre, int maxBooks) {
        String QUERY = "SELECT id, title, year, num_pages FROM table WHERE title LIKE \"%" + genre + \"%\";\n        int COLUMN_INDEX_ID = 0;
        int COLUMN_INDEX_TITLE = 1;
        int COLUMN_INDEX_YEAR = 2;
        int COLUMN_INDEX_NUM_PAGES = 3;
        Database database = new Database("lou", "PA$$W0RD", "https://acme-books.com/db");
        Cursor cursor = database.cursor();
        Booklist booklist = new Booklist();
        List titles = new ArrayList();
        try {
            cursor.execute(QUERY);
            boolean finished = false;
            if (cursor.rowCount() > 0) {
                int rowNumber = 0;
                while (!finished) {
                    int rowCount = cursor.rowCount();
                    for (int i = 0; i < Math.min(rowCount, maxBooks); ++i) {
                        cursor.fetchone();
                        int id = cursor.getInt(COLUMN_INDEX_ID);
                        String title = cursor.getString(COLUMN_INDEX_TITLE);
                        int year = cursor.getInt(COLUMN_INDEX_YEAR);
                        int num_pages = cursor.getInt(COLUMN_INDEX_NUM_PAGES);
                        Book book = new Book(id, title, year, num_pages);
                        if (title != null) {
                            titles.add(title);
                        }
                        if (id != -1) {
                            boolean bestseller = isBestseller(book.getId());
                            if (bestseller) {
                                booklist.hasBestseller = bestseller;
                                booklist.addBook(book);
                            }
                        }
                    }
                    if (DEBUG) {
                        System.out.println("Fetched book: "+ title + " (" + genre + "+")");
                    }
                }
            }
            return booklist;
        } finally {
            if (finished) {
                System.out.println("No books found.");
            }
        }
    }
}

public boolean isBestseller(int id) {
    // Implementation of isBestseller method
    return false;
}

public class Book {
    public Book(int id, String title, int year, int num_pages) {
        // Constructor
    }

    public void addBook(Book book) {
        // Add book to collection
    }

    public List getBooks() {
        // Get list of books
        return null;
    }
}

public class Booklist {
    public List titles;
    public boolean hasBestseller;
    public void addBook(Book book) {
        // Add book to collection
    }
    public List getBooks() {
        // Get list of books
        return null;
    }
}

public class Database {
    public Cursor cursor;
    public Database(String username, String password, String url) {
        // Constructor
    }
}

public class Cursor {
    public int rowCount() {
        // Returns row count
        return 0;
    }
    public void fetchone() {
        // Fetch one row
    }
    public String getString(int columnIndex) {
        // Get string value
        return null;
    }
    public int getInt(int columnIndex) {
        // Get integer value
        return 0;
    }
}

public class BookDatabase {
    public static void main(String[] args) {
        // Main method
    }
}
rowNumber++;

if (cursor.end() || rowNumber >= maxBooks) {
    finished = true;
} else if (i == rowCount - 1) {
    cursor.next(id);
}

} else {
    System.out.println("No results found in the database");
}

} catch (ConnectionException exception) {
    exception.printStackTrace();
}

booklist.titles = titles;
return booklist;

}

public boolean isBestseller(int id) {
    return id == 2;
}

public static void main(String[] args) {
    new BookListing().getBookListing("romance", 3);
}
import org.acme.database.Database;
import org.acme.database.Cursor;
import org.acme.database.Book;
import org.acme.database.ConnectionException;

public class ExtractedExample {

  public static void main(String[] args) {
    String QUERY = "SELECT id, title, year, num_pages FROM table WHERE title LIKE '%" + "romance" + '%";
    Database database = new Database("lou", "PA$$W0RD", "https://acme-books.com/db");
    Cursor cursor = database.cursor();
    try {
      cursor.execute(QUERY);
      if (cursor.rowCount() > 0) {
        int id = cursor.getInt(0);
        String title = cursor.getString(1);
        int year = cursor.getInt(2);
        int num_pages = cursor.getInt(3);
        Book book = new Book(id, title, year, num_pages);
        System.out.println("Fetched book: " + title + " (" + "romance" + ")");
      }
    } catch (ConnectionException exception) {
    }
  }
}
public class ExtractedExample {

    public static void main(String[] args) throws ConnectionException {

        Database database = new Database("lou", "PA$$W0RD", "https://acme-books.com/db");
        Cursor cursor = database.cursor();
        try {
            cursor.execute("SELECT id, title, year, num_pages FROM table WHERE title LIKE '%romance\%';");
            if (cursor.rowCount() > 0) {
                int rowCount = cursor.rowCount();
                for (int i = 0; i < Math.min(rowCount, 3); ++i) {
                    cursor.fetchone();
                    int id = cursor.getInt(0);
                    String title = cursor.getString(1);
                }
            }
        } catch (ConnectionException exception) {
        }
    }
}
public class ExtractedExample {

    public static void main(String[] args) throws ConnectionException {

        String QUERY = "SELECT id, title, year, num_pages FROM table WHERE title LIKE '%" + "romance" + '%";

        Database database = new Database("lou", "PA$$W0RD", "https://acme-books.com/db");

        Cursor cursor = database.cursor();

        cursor.execute(QUERY);
        cursor.fetchone();

        String title = cursor.getString(1);
        int year = cursor.getInt(2);
        int num_pages = cursor.getInt(3);

        Book book = new Book(1, title, year, num_pages);

        if (true == true) {
            System.out.println("Fetched book: " + title + " (" + "romance" + ")");
        }
    }
}

A5. Source Programs and Examples
Task 1, Participant 15

Buggy: While the participant finished extracting the example, it does not compile and run: the participant’s first selection was a return statement that conflicts with the main function’s signature, and which could only be removed if they started over.

```java
import org.acme.database.Booklist;
import org.acme.database.Database;
import org.acme.database.Cursor;
import org.acme.database.ConnectionException;
import org.acme.database.Book;

public class ExtractedExample {

  public static void main(String[] args) throws ConnectionException {

    String QUERY = "SELECT id, title, year, num_pages FROM table WHERE title LIKE '\'%\' + "romance" + \'%'";
    Database database = new Database("lou", "PA$$W0RD", "https://acme-books.com/db");
    Cursor cursor = database.cursor();
    Booklist booklist = new Booklist();

    cursor.execute(QUERY);
    boolean finished = false;
    if (cursor.rowCount() > 0) {
      while (finished == false) {
        int rowCount = cursor.rowCount();
        cursor.fetchone();
        int id = cursor.getInt(0);
        String title = cursor.getString(1);
        int year = cursor.getInt(2);
        int num_pages = cursor.getInt(3);
        Book book = new Book(id, title, year, num_pages);
        finished = true;
      }
    }
    return book;
  }
}
```

A5. Source Programs and Examples
public class ExtractedExample {

  public static void main(String[] args) throws ConnectionException {

    String QUERY = "SELECT id, title, year, num_pages FROM table WHERE title LIKE '%\"romance\"'";

    Database database = new Database("lou", "PA$$W0RD", "https://acme-books.com/db");

    Cursor cursor = database.cursor();

    try {
      cursor.execute(QUERY);
      if (cursor.rowCount() > 0) {
        int rowCount = cursor.rowCount();
        cursor.fetchone();
      }
    } catch (ConnectionException exception) {

    }

  }

}
public class ExtractedExample {

    public static void main(String[] args) throws ConnectionException {

        int COLUMN_INDEX_ID = 0;
        int COLUMN_INDEX_TITLE = 1;
        int COLUMN_INDEX_YEAR = 2;
        int COLUMN_INDEX_NUM_PAGES = 3;

        Database database = new Database("lou", "PA$$W0RD", "https://acme-books.com/db");

        Cursor cursor = database.cursor();

        cursor.execute("SELECT id, title, year, num_pages FROM table WHERE title LIKE '%romance%'");

        cursor.fetchone();

        int id = cursor.getInt(COLUMN_INDEX_ID);
        String title = cursor.getString(COLUMN_INDEX_TITLE);
        int year = cursor.getInt(COLUMN_INDEX_YEAR);
        int num_pages = cursor.getInt(COLUMN_INDEX_NUM_PAGES);

        Book book = new Book(id, title, year, num_pages);

        System.out.println(title);
    }
}
import com.sun.mail.smtp.SMTPTransport;
import org.apache.commons.lang.Validate;
import org.apache.commons.lang.StringUtils;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.Jsoup;
import org.jsoup.select.Elements;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.net.URL;
import java.net.URLEncoder;
import java.security.Security;
import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Properties;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class CraigslistMonitor {

    public static void main(String[] args) throws IOException, MessagingException, AddressException {
        String arg0 = "electric bike";
        String arg1 = "openairandrew@gmail.com";

        List titles = new ArrayList();
        List prices = new ArrayList();
        List links = new ArrayList();

        String query = URLEncoder.encode(arg0, "utf-8");
        String destination = arg1;
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);
        System.out.println("Fetching " + url.toExternalForm() + "...");

        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");
        System.out.println(url);

        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();)
            try {
                Element row = (Element) rowsIterator.next();
                String price = row.select("span.result-meta span.result-price").text();
                int priceInt;

                priceInt = Integer.parseInt(price.replaceFirst("\$", "");

                catch (NumberFormatException exception) {
                    priceInt = -1;
        
    public static void main(String[] args) throws IOException, MessagingException, AddressException {
        String arg0 = "electric bike";
        String arg1 = "openairandrew@gmail.com";

        List titles = new ArrayList();
        List prices = new ArrayList();
        List links = new ArrayList();

        String query = URLEncoder.encode(arg0, "utf-8");
        String destination = arg1;
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);
        System.out.println("Fetching " + url.toExternalForm() + "...");

        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");
        System.out.println(url);

        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();)
            try {
                Element row = (Element) rowsIterator.next();
                String price = row.select("span.result-meta span.result-price").text();
                int priceInt;

                priceInt = Integer.parseInt(price.replaceFirst("\$", "");

                catch (NumberFormatException exception) {
                    priceInt = -1;
```java
prices.add(new Integer(priceInt));

Element image = row.select("a.result-image.gallery").get(0);
String link = image.attr("abs:href");
links.add(link);

String title = row.select("a.result-title.hdrlnk").text();
titles.add(title);

if (titles.size() >= 5) break;
}

int maxTitleLength = 0;
int maxPriceLength = 0;
int maxLinkLength = 0;
for (int i = 0; i < titles.size(); i++) {
    String titleText = (String) titles.get(i);
    if (titleText.length() > maxTitleLength) {
        maxTitleLength = titleText.length();
    }
    int priceLine = ((Integer) prices.get(i)).intValue();
    if (Integer.toString(priceLine).length() > maxPriceLength) {
        maxPriceLength = Integer.toString(priceLine).length();
    }
    String linkText = (String) links.get(i);
    if (linkText.length() > maxLinkLength) {
        maxLinkLength = linkText.length();
    }
}

String messageHtml = "<code>
for (int j = 0; j < titles.size(); j++) {
    String titleText = (String) titles.get(j);
    String priceText = ((Integer) prices.get(j)).toString();
    String linkText = (String) links.get(j);
    String titlePadded = titleText + StringUtils.repeat(" ", maxTitleLength - titleText.length());
    String pricePadded = priceText + StringUtils.repeat(" ", maxPriceLength - priceText.length());
    String linkPadded = linkText + StringUtils.repeat(" ", maxLinkLength - linkText.length());
    messageHtml += ("( $" + pricePadded + ")" + titlePadded + "<br/>" + linkPadded + "<br/><br/>");
}
messageHtml += "</code>

BufferedReader confReader = new BufferedReader(new FileReader("/etc/smtp.conf"));
String username = confReader.readLine();
String password = confReader.readLine();
System.out.println("Logging in with " + username + ", " + password);

Properties properties = System.getProperties();
properties.setProperty("mail.smtps.host", "smtp.gmail.com");
properties.setProperty("mail.smtps.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
properties.setProperty("mail.smtps.socketFactory.port", "587");
properties.setProperty("mail.smtps.quitwait", "false");
session = Session.getInstance(properties, null);
MimeMessage message = new MimeMessage(session);
message.setFrom(new InternetAddress(username));
```

A5. Source Programs and Examples
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destination, false));
message.setSubject("Update of Craigslist posts");
message.setText(messageHtml, "utf-8", "html");
message.setSentDate(new Date());

SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");
transport.connect("smtp.gmail.com", username, password);
transport.sendMessage(message, message.getAllRecipients());
transport.close();
}
public class ExtractedExample {

    public static void main(String[] args) throws IOException, java.lang.NumberFormatException {

        int priceInt;

        String arg0 = "electric bike";
        List prices = new ArrayList();
        String query = URLEncoder.encode(arg0, "utf-8");
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);

        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");
        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();)
        
            Element row = (Element) rowsIterator.next();

            String price = row.select("span.result-meta span.result-price").text();
            priceInt = Integer.parseInt(price.replaceFirst("\$", ")");

            prices.add(new Integer(priceInt));

        System.out.println(prices);
    }
}

Exemplar: Creating an example primarily through manual selections, with a little bit of assistance from CodeScoop.
import java.util.List;
import java.util.ArrayList;
import java.net.URLEncoder;
import java.net.URL;
import org.jsoup.nodes.Document;
import org.jsoup.Jsoup;
import org.jsoup.select.Elements;
import java.util.Iterator;
import org.jsoup.nodes.Element;
import java.io.IOException;

public class ExtractedExample {

    public static void main(String[] args) throws IOException, java.lang.NumberFormatException {
        int priceInt;

        String arg0 = "electric bike";
        List prices = new ArrayList();

        String query = URLEncoder.encode(arg0, "utf-8");
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);
        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");

        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();) {
            Element row = (Element) rowsIterator.next();
            String price = row.select("span.result-meta span.result-price").text();
            try {
                priceInt = Integer.parseInt(price.replaceFirst("\$", ""));
            } catch (NumberFormatException exception) {
                priceInt = -1;
            }
            prices.add(new Integer(priceInt));
        }
        System.out.println(prices);
    }
}
Exemplar: The author makes an initial selection, and the rest of the example is completed automatically by CodeScoop.

```java

public class ExtractedExample {

    public static void main(String[] args) throws IOException {

        List titles = new ArrayList();
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=electric+bike");
        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");
        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();) {
            Element row = (Element) rowsIterator.next();
            String title = row.select("a.result-title.hdrlnk").text();
            titles.add(title);
        }
    }
}
```
Task 2, Participant 8

Buggy: After seeing their code execute, Participant 8’s last step was to add the erroneous print statement on line 32 (buggy because price is out of scope). While their code did not compile with the addition of the print statement, we count this example as “finished” because the program had been verified to run correctly just before, and the participant acknowledged the error.

```java
import java.net.URLEncoder;
import java.net.URL;
import org.jsoup.nodes.Document;
import org.jsoup.Jsoup;
import org.jsoup.select.Elements;
import java.util.Iterator;
import org.jsoup.nodes.Element;
import java.io.IOException;
public class ExtractedExample {
    public static void main(String[] args) throws IOException {
        String arg0 = "electric bike";
        String query = URLEncoder.encode(arg0, "utf-8");
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);
        System.out.println("Fetching " + url.toExternalForm() + "...");
        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");
        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();) {
            Element row = (Element) rowsIterator.next();
            String price = row.select("span.result-meta span.result-price").text();
            System.out.println(price);
        }
    }
}
```

A5. Source Programs and Examples
import java.net.URL;
import org.jsoup.nodes.Document;
import org.jsoup.Jsoup;
import org.jsoup.select.Elements;
import java.util.Iterator;
import org.jsoup.nodes.Element;
import java.io.IOException;

public class ExtractedExample {
    public static void main(String[] args) throws IOException {
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + "electric+bike");
        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");
        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();)
            
        }
    }
}
import java.util.List;
import java.util.ArrayList;
import java.net.URLEncoder;
import java.net.URL;
import org.jsoup.nodes.Document;
import org.jsoup.Jsoup;
import org.jsoup.select.Elements;
import java.util.Iterator;
import org.jsoup.nodes.Element;
import java.io.IOException;

public class ExtractedExample {

    public static void main(String[] args)
        throws IOException {

        int priceInt;

        List titles = new ArrayList();
        List prices = new ArrayList();
        List links = new ArrayList();

        String query = URLEncoder.encode("electric bike", "utf-8");
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);
        Document doc = Jsoup.parse(url, 3*1000);

        Elements rows = doc.select("li.result-row");

        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();)
            { 
            
            Element row = (Element) rowsIterator.next();

            String price = row.select("span.result-meta span.result-price").text();
            try { 
                priceInt = -1;
            } 
            catch (NumberFormatException exception) {
                priceInt = -1;
            }
            prices.add(new Integer(priceInt));
            
            Element image = row.select("a.result-image.gallery").get(0);
            String link = image.attr("abs:href");
            links.add(link);

            String title = row.select("a.result-title.hdrlnk").text();
            titles.add(title);

            if (titles.size() >= 5) break;
            } 

        for (int j = 0; j < titles.size(); j++)
            { 
            
            String titleText = (String) titles.get(j);
            String priceText = ((Integer) prices.get(j)).toString();
            String linkText = (String) links.get(j);

            String titlePadded = titleText + StringUtils.repeat(" ", maxTitleLength - titleText.length());
            String pricePadded = priceText + StringUtils.repeat(" ", maxPriceLength - priceText.length());
            String linkPadded = linkText + StringUtils.repeat(" ", maxLinkLength - linkText.length());

            messageHtml += (( $" + pricePadded + " ) + titlePadded + "<br/>" + linkPadded + "<br/><br/>");
            } 

        messageHtml += "</code>";
    }
}

A5. Source Programs and Examples

Task 2, Participant 16

Incomplete: This participant did not finish the example because they ran out of time.
import java.net.URLEncoder;  
import java.net.URL;  
import org.jsoup.nodes.Document;  
import org.jsoup.Jsoup;  
import org.jsoup.select.Elements;  
import java.util.Iterator;  
import org.jsoup.nodes.Element;  
import java.io.IOException;  

public class ExtractedExample {  

    public static void main(String[] args) throws IOException {  
        String arg0 = "electric bike";  
        String query = URLEncoder.encode(arg0, "utf-8");  
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);  
        System.out.println("Fetching " + url.toExternalForm() + "...");  
        Document doc = Jsoup.parse(url, 3*1000);  
        Elements rows = doc.select("li.result-row");  
        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();) {  
            Element row = (Element) rowsIterator.next();  
            String price = row.select("span.result-meta span.result-price").text();  
        }  
    }  
}
import com.sun.mail.smtp.SMTPTransport;
import org.apache.commons.lang3.Validate;
import org.apache.commons.lang3.StringUtils;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.Jsoup;
import org.jsoup.select.Elements;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.net.URL;
import java.net.URLEncoder;
import java.security.Security;
import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Properties;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

*/
public class CraigslistMonitor {

    public static void main(String[] args) throws IOException, MessagingException, AddressException {

        String arg0 = "electric bike";
        String arg1 = "openairandrew@gmail.com";

        List titles = new ArrayList();
        List prices = new ArrayList();
        List links = new ArrayList();

        String query = URLEncoder.encode(arg0, "utf-8");
        String destination = arg1;
        URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);
        System.out.println("Fetching " + url.toExternalForm() + "...");

        Document doc = Jsoup.parse(url, 3*1000);
        Elements rows = doc.select("li.result-row");
        System.out.println(url);

        for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();) {
            Element row = (Element) rowsIterator.next();

            String price = row.select("span.result-meta span.result-price").text();
            int priceInt;
            try {
                priceInt = Integer.parseInt(price.replaceFirst("\$", ""));
            } catch (NumberFormatException exception) {
                priceInt = -1;
            }

            System.out.println("Title: " + row.select("a.result-title").text());
            System.out.println("Price: " + priceInt);
            System.out.println("Destination: " + destination);
            System.out.println("URL: " + url);
            System.out.println("For sale: " + row.select("span.result-meta span.result-price").text());
            System.out.println("Location: " + row.select("span.result-meta span.result-location").text());
        }
    }
}
Element image = row.select("a.result-image.gallery").get(0);
String link = image.attr("abs:href");
links.add(link);

String title = row.select("a.result-title.hdrlnk").text();
titles.add(title);
if (titles.size() >= 5) break;
}

int maxTitleLength = 0;
int maxPriceLength = 0;
int maxLinkLength = 0;
for (int i = 0; i < titles.size(); i++) {
    String titleLine = (String) titles.get(i);
    if (titleLine.length() > maxTitleLength) {
        maxTitleLength = titleLine.length();
    }
    int priceLine = ((Integer) prices.get(i)).intValue();
    if (Integer.toString(priceLine).length() > maxPriceLength) {
        maxPriceLength = Integer.toString(priceLine).length();
    }
    String linkLine = (String) links.get(i);
    if (linkLine.length() > maxLinkLength) {
        maxLinkLength = linkLine.length();
    }
}

String messageHtml = "<code>
for (int j = 0; j < titles.size(); j++) {
    String titleText = (String) titles.get(j);
    String priceText = ((Integer) prices.get(j)).toString();
    String linkText = (String) links.get(j);
    String titlePadded = titleText + StringUtils.repeat(" ", maxTitleLength - titleText.length());
    String pricePadded = priceText + StringUtils.repeat(" ", maxPriceLength - priceText.length());
    String linkPadded = linkText + StringUtils.repeat(" ", maxLinkLength - linkText.length());
    messageHtml += ("( \$\" + pricePadded + \"") + titlePadded + "<br/>" + linkPadded + "<br/>"");
}
messageHtml += "</code>
BufferedReader confReader = new BufferedReader(new FileReader("/etc/smtp.conf"));
String username = confReader.readLine();
String password = confReader.readLine();
System.out.println("Logging in with " + username + ", " + password);

Properties properties = System.getProperties();
properties.setProperty("mail.smtps.host", "smtp.gmail.com");
properties.setProperty("mail.smtps.socketFactory.class", "SSLSocketFactory");
properties.setProperty("mail.smtps.port", "587");
properties.setProperty("mail.smtps.socketFactory.port", "587");
properties.setProperty("mail.smtp.starttls.enable", "true");
properties.put("mail.smtp.quitwait", "false");
Session session = Session.getInstance(properties, null);

MimeMessage message = new MimeMessage(session);
message.setFrom(new InternetAddress(username));
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destination, false));
message.setSubject("Update of Craigslist posts");
message.setText(messageHtml, "utf-8", "html");
message.setSentDate(new Date());

SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");
transport.connect("smtp.gmail.com", username, password);
transport.sendMessage(message, message.getAllRecipients());
transport.close();
}
public class ExtractedExample {  
   
   public static void main(String[] args) throws MessagingException, IOException, java.lang.NumberFormatException {
      
      String query = URLEncoder.encode(args[0], "utf-8");
      String destination = args[1];
      URL url = new URL("https://sfbay.craigslist.org/search/bia?query=" + query);
      
      Document doc = Jsoup.parse(url, 3*1000);
      Elements rows = doc.select("li.result-row");
      for (Iterator rowsIterator = rows.iterator(); rowsIterator.hasNext();) {
         Element row = (Element) rowsIterator.next();
         String price = row.select("span.result-meta span.result-price").text();
         if (!price.equals("")) {
            priceInt = Integer.parseInt(price.replaceFirst("\$", ""));
         }
         prices.add(new Integer(priceInt));
         Element image = row.select("a.result-image.gallery").get(0);
         String link = image.attr("abs:href");
         links.add(link);
         String title = row.select("a.result-title.hdrlnk").text();
         titles.add(title);
         if (titles.size() >= 5) break;
      }
      
      int maxTitleLength = 0;
      int maxPriceLength = 0;
      for (Integer int i = 0; i < prices.size(); i++) {
         if (priceLine > maxPriceLength) {
            maxPriceLength = priceLine;
         }
      }
      
      String messageHtml = "<code>";
      for (int j = 0; j < titles.size(); j++) {
         String titleText = (String) titles.get(j);
         String priceText = ((Integer) prices.get(j)).toString();
         String linkText = (String) links.get(j);
         String titlePadded = titleText + StringUtils.repeat(" ", maxTitleLength - titleText.length());
         String pricePadded = priceText + " ", maxPriceLength - priceText.length());
         String linkPadded = linkText + StringUtils.repeat(" ", maxLinkLength - linkText.length());
         
      }  
   }
}

Exemplar: An author on "auto-pilot": they accepted all code suggestions that CodeScoop made.
The resulting example is essentially the same as a conservative program slice.

Incomplete: Participant 1 did not finish the example, because they encountered a bug with the tool. After Participant 1, the Task 3 source program was updated so other participants would not encounter the same bug. Lines 15 and 30 in this example come from a previous version of the task 3 program.
BufferedReader confReader = new BufferedReader(new FileReader("/etc/smtp.conf"));

String username = confReader.readLine();
String password = confReader.readLine();

System.out.println("Logging in with " + username + ", " + password);

String sslFactoryClass = "javax.net.ssl.SSLSocketFactory";

Properties properties = System.getProperties();

properties.setProperty("mail.smtps.host", "smtp.gmail.com");
properties.setProperty("mail.smtp.socketFactory.class", sslFactoryClass);
properties.setProperty("mail.smtp.port", "587");
properties.setProperty("mail.smtp.socketFactory.port", "587");
properties.setProperty("auth", "true");

properties.put("mail.smtps.quitwait", "false");

Session session = Session.getInstance(properties, null);

MimeMessage message = new MimeMessage(session);
message.setFrom(new InternetAddress(username));
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destination, false));
message.setSubject("Update of Craigslist posts");
message.setText(messageHtml, "utf-8", "html");
message.setSentDate(new Date());

SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");
transport.connect("smtp.gmail.com", username, password);
transport.sendMessage(message, message.getAllRecipients());
public class ExtractedExample {
  public static void main(String[] args) throws MessagingException, IOException {
    String arg1 = "openairandrew@gmail.com";
    String destination = arg1;
    String messageHtml = "<code>";

    BufferedReader confReader = new BufferedReader(new FileReader("/etc/smtp.conf"));
    String username = confReader.readLine();
    String password = confReader.readLine();
    String sslFactoryClass = "javax.net.ssl.SSLSocketFactory";

    Properties properties = System.getProperties();
    properties.setProperty("mail.smtps.host", "smtp.gmail.com");
    properties.setProperty("mail.smtps.socketFactory.class", sslFactoryClass);
    properties.setProperty("mail.smtps.port", "587");
    properties.setProperty("mail.smtps.socketFactory.port", "587");
    properties.setProperty("auth", "true");
    properties.put("mail.smtps.quitwait", "false");

    Session session = Session.getInstance(properties, null);

    MimeMessage message = new MimeMessage(session);
    message.setFrom(new InternetAddress(username));
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destination, false));
    message.setSubject("Update of Craigslist posts");
    message.setText(messageHtml, "utf-8", "html");
    message.setSentDate(new Date());

    SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");
    transport.connect("smtp.gmail.com", username, password);
    transport.sendMessage(message, message.getAllRecipients());
  }
}
public class ExtractedExample {
    public static void main(String[] args) throws MessagingException, IOException {
        String arg1 = "openairandrew@gmail.com";
        String destination = arg1;
        String messageHtml = "<code>";
        BufferedReader confReader = new BufferedReader(new FileReader("/etc/smtp.conf"));
        String username = confReader.readLine();
        String password = confReader.readLine();
        String sslFactoryClass = "javax.net.ssl.SSLSocketFactory";
        Properties properties = System.getProperties();
        properties.setProperty("mail.smtps.host", "smtp.gmail.com");
        properties.setProperty("mail.smtp.socketFactory.class", sslFactoryClass);
        properties.setProperty("mail.smtp.port", "587");
        properties.setProperty("mail.smtp.socketFactory.port", "587");
        properties.put("mail.smtps.quitwait", "false");
        Session session = Session.getInstance(properties, null);
        MimeMessage message = new MimeMessage(session);
        message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destination, false));
        message.setText(messageHtml, "utf-8", "html");
        SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");
        transport.connect("smtp.gmail.com", username, password);
        transport.sendMessage(message, message.getAllRecipients());
        transport.close();
    }
}
public class ExtractedExample {

    String sslFactoryClass = "javax.net.ssl.SSLSocketFactory";

    Properties properties = System.getProperties();
    properties.setProperty("mail.smtps.host", "smtp.gmail.com");
    properties.setProperty("mail.smtp.socketFactory.class", sslFactoryClass);
    properties.setProperty("mail.smtp.port", "587");
    properties.setProperty("mail.smtp.socketFactory.port", "587");
    properties.setProperty("auth", "true");
    properties.put("mail.smtps.quitwait", "false");

    Session session = Session.getInstance(properties, null);
    MimeMessage message = new MimeMessage(session);
    message.setFrom(new InternetAddress("openairandrew@gmail.com"));
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse("openairandrew@gmail.com", false));
    message.setSubject("Update of Craigslist posts");
    message.setSentDate(new Date());

    SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");
    transport.connect("smtp.gmail.com", "openairandrew@gmail.com", "password");
    transport.sendMessage(message, message.getAllRecipients());
    transport.close();

    }
}
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.Properties;
import javax.mail.Session;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.InternetAddress;
import javax.mail.Message;
import java.util.Date;
import com.sun.mail.smtp.SMTPTransport;
import javax.mail.MessagingException;
import java.io.IOException;
import java.security.Security;

public class ExtractedExample {
    public static void main(String[] args) throws MessagingException, IOException {
        String arg1 = "openairandrew@gmail.com";
        String destination = arg1;

        BufferedReader confReader = new BufferedReader(new FileReader("/etc/smtp.conf"));
        String username = confReader.readLine();
        String password = confReader.readLine();

        System.out.println("Logging in with ", username + ", " + password);

        Security.addProvider(new com.sun.net.ssl.internal.ssl.Provider());
        String sslFactoryClass = "javax.net.ssl.SSLSocketFactory";

        Properties properties = System.getProperties();
        properties.setProperty("mail.smtps.host", "smtp.gmail.com");
        properties.setProperty("mail.smtp.socketFactory.class", sslFactoryClass);
        properties.setProperty("mail.smtp.port", "587");

        SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");

        transport.connect("smtp.gmail.com", username, password);

        transport.sendMessage(message, message.getAllRecipients());
        transport.close();
    }
}

A5. Source Programs and Examples
```java
public class ExtractedExample {

    public static void main(String[] args) throws MessagingException {

        String destination = "openairandrew@gmail.com";
        String messageHtml = "<code>";

        Properties properties = System.getProperties();

        properties.setProperty("mail.smtps.host", "smtp.gmail.com");
        properties.setProperty("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        properties.setProperty("mail.smtp.port", "587");
        properties.setProperty("mail.smtp.socketFactory.port", "587");
        properties.setProperty("auth", "true");
        properties.put("mail.smtps.quitwait", "false");

        Session session = Session.getInstance(properties, null);

        MimeMessage message = new MimeMessage(session);

        message.setFrom(new InternetAddress("openairandrew@gmail.com"));
        message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destination, false));
        message.setSubject("Update of Craigslist posts");
        message.setText(messageHtml, "utf-8", "html");
        message.setSentDate(new Date());

        SMTPTransport transport = (SMTPTransport) session.getTransport("smtps");

        transport.connect("smtp.gmail.com", "openairandrew@gmail.com", "<password>");

        transport.sendMessage(message, message.getAllRecipients());
    }
}
```