

# Applying AI+ to Review Programming Assignment Submissions in Vandy CS Courses

**Douglas C. Schmidt**

**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

**[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)**

**Professor of Computer Science**

**Institute for Software  
Integrated Systems**

**Vanderbilt University  
Nashville, Tennessee, USA**



# Learning Objectives in this Part of the Lesson

- Understand the difference between generative artificial intelligence & augmented intelligence (AI+)
- Learn how we're applying AI+ to enhance lecture material & quiz/exam assessments in Vandy CS courses
- Recognize how ChatGPT-4 can automate key steps in reviewing programming assignment solutions

## Question 1: Trace the `compute()` method

1. Inspecting the `compute()` method line-by-line.
2. No direct call to `submit()` method is observed.

## Question 2: Methods Accessing `mMaxDepth`

1. Inspecting each method line-by-line.
2. No methods directly access the `mMaxDepth` field.

## Question 3: Try/Catch Blocks

1. Inspecting each method line-by-line.
2. No try/catch blocks are explicitly used in the provided Java code.

## Question 4: Use of AtomicReference or AtomicInteger

1. Inspecting each method line-by-line.
2. No usage of AtomicReference or AtomicInteger is observed.

## Question 5: For Loops

1. Inspecting each method line-by-line.
2. No for loops meeting the specified criteria are found.

## Question 6: Direct Calls to `callInManagedBlocker()`

1. Inspecting each method line-by-line.
2. No methods directly call `callInManagedBlocker()`.

# Using AI+ to Review Programming Assignment Submissions

---


- AI+ helps improve reviewing of programming assignment submissions for my CS courses



# Using AI+ to Review Programming Assignment Submissions

- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!

D

Please read the following Java code skeleton and generate a complete solution that uses the Java fork-join framework to perform web crawling concurrently. 

```
package edu.vanderbilt.imagecrawler.crawlers;

import static edu.vanderbilt.imagecrawler.crawlers.Crawler.Type.IMAGE;
import static edu.vanderbilt.imagecrawler.crawlers.Crawler.Type.PAGE;

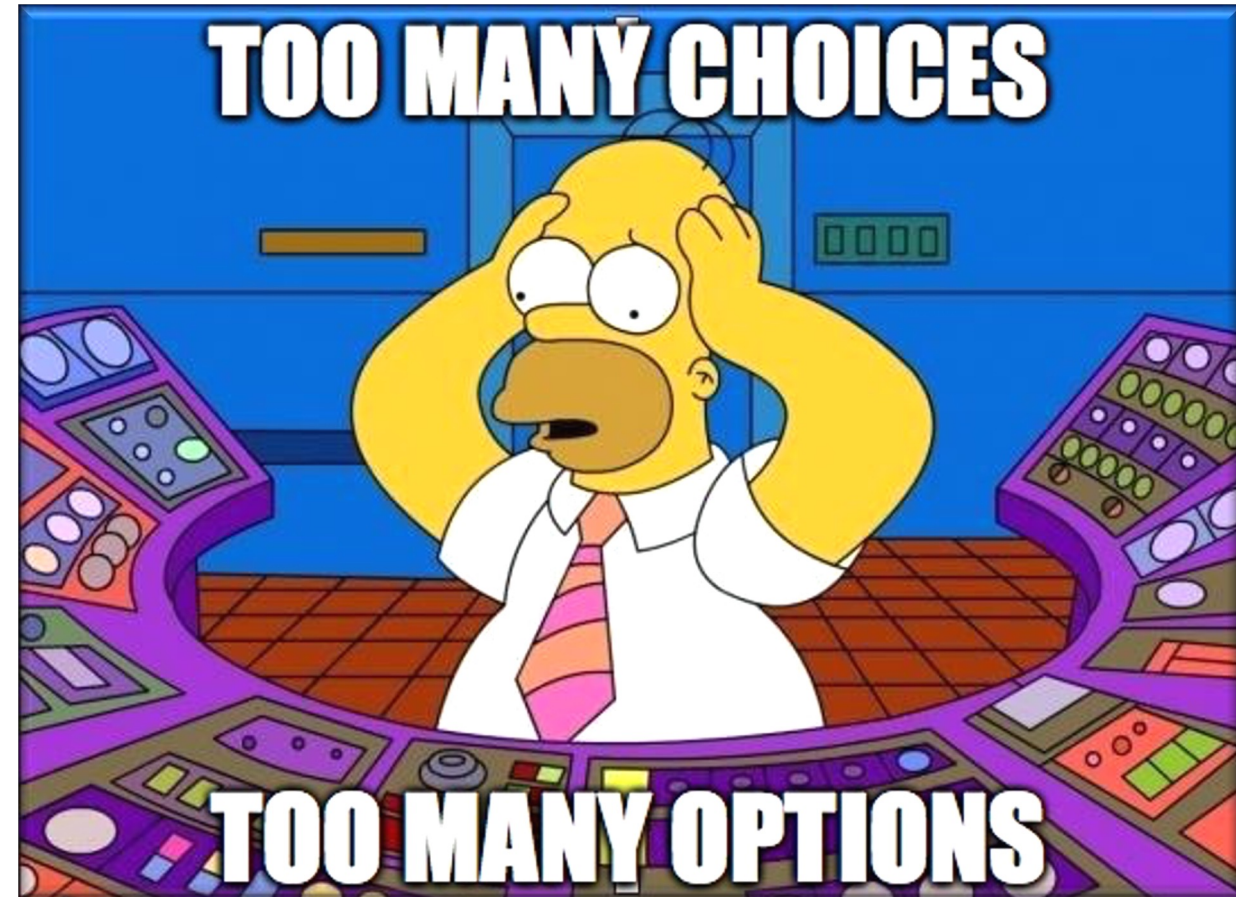
import java.net.URL;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.ForkJoinPool;
import java.util.concurrent.ForkJoinTask;
import java.util.concurrent.RecursiveTask;
import java.util.function.Function;
import java.util.function.Supplier;

import edu.vanderbilt.imagecrawler.platform.Cache;
import edu.vanderbilt.imagecrawler.transforms.Transform;
import edu.vanderbilt.imagecrawler.utils.BlockingTask;
import edu.vanderbilt.imagecrawler.utils.ExceptionUtils;
import edu.vanderbilt.imagecrawler.utils.Image;
import edu.vanderbilt.imagecrawler.web.WebPageElement;

/**
 * This {@link ImageCrawler} implementation strategy uses the Java
 * common fork-join framework to perform an "image crawl" starting
 * from a root Uri. Images from an HTML page reachable from the root
 * Uri are downloaded in parallel from a remote web server or the
 * local file system, transformed, and then stored in files on the
 * Android device, where they can be displayed to the user. This
 * implementation should not use any Java streams features.
 */
public class ForkJoinCrawler
```

# Using AI+ to Review Programming Assignment Submissions

- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!
  - But this creates a whole new challenge for assessing programming submissions!





# Using AI+ to Review Programming Assignment Submissions

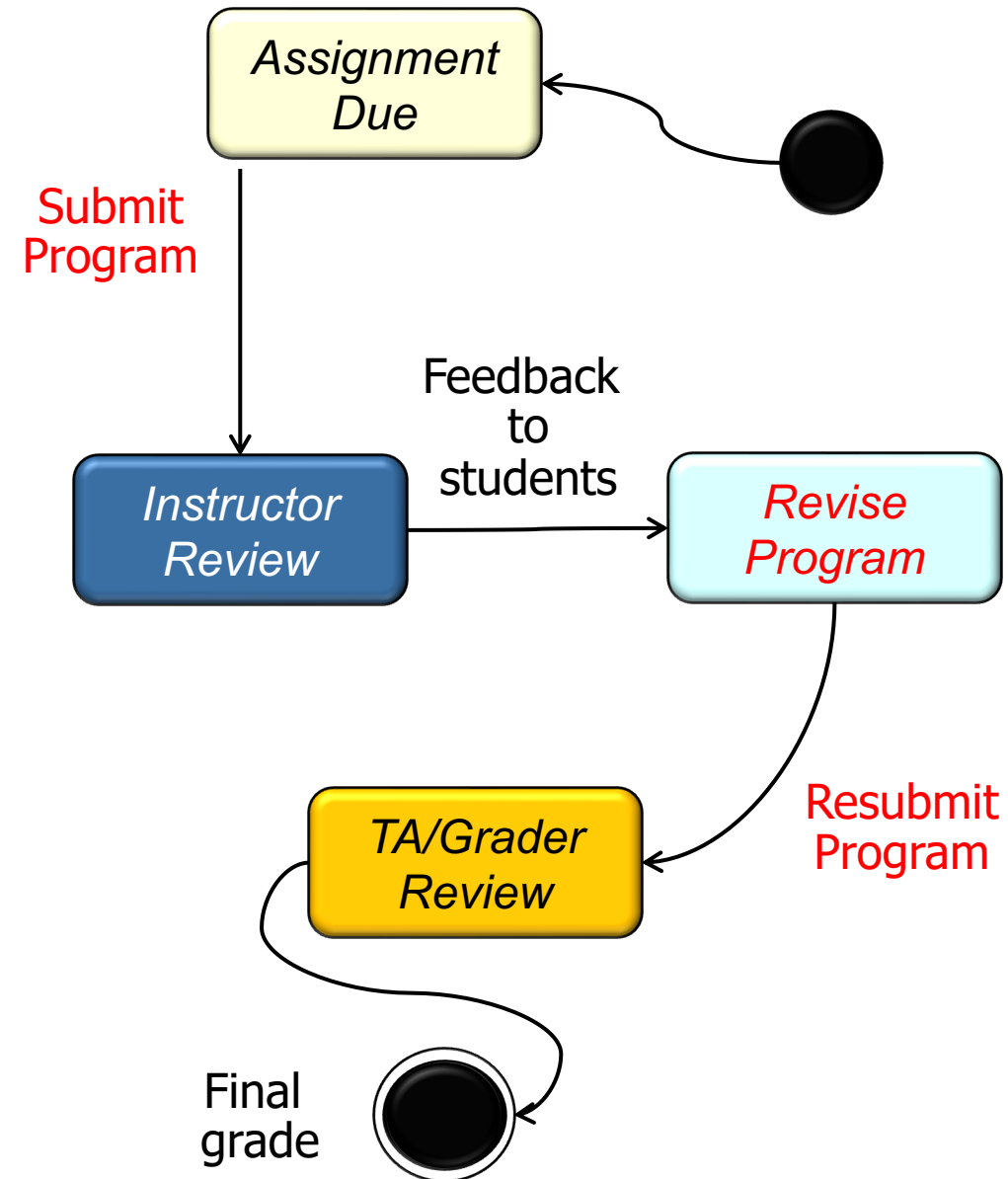
- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!
    - But this creates a whole new challenge for assessing programming submissions!
      - Hard to check automatically via white-box testing & "mocking"

```
@Test
fun `makeProcessImageTask() has the correct solution`() {
    mockkStatic(ExceptionUtils::class)
    val url = mockk<URL>()
    val f = mockk<Function<String, URL>>()
    every { ExceptionUtils.rethrowFunction<String, URL>(any()) } returns f
    every { f.apply(any()) } returns url
    val result = crawler.makeProcessImageTask( url: "mock")
    assertThat(result).isInstanceOf(ProcessImageTask::class.java)
    assertThat(result.value<URL>(URL::class.java)).isSameAs(url)
    verify(exactly = 1) { this: MockKVerificationScope
        f.apply(any())
        ExceptionUtils.rethrowFunction<String, URL>(any())
    }
}

@Test
fun `makeURLCrawlerTest() has the correct solution`() {
    mockkConstructor(ForkJoinCrawler.URLCrawlerTask::class)
    val depth = -99
    val result = crawler.makeURLCrawlerTask( pageUri: "mock", depth)
    isMockKMock(result)
}
```

# Using AI+ to Review Programming Assignment Submissions

- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!
- Automating key steps in reviewing programming assignment solutions



# Using AI+ to Review Programming Assignment Submissions

- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!
- Automating key steps in reviewing programming assignment solutions

```
protected ForkJoinTask<Integer> makeProcessImageTask(String url) {
    // @@ Replace this try/catch block with a call to
    // ExceptionUtils.rethrowFunction().
    Function<String, URL> urlConverter = urlStr -> {
        try {
            return new URL(urlStr);
        } catch (Exception e) {
            throw new RuntimeException("...");
        }
    };

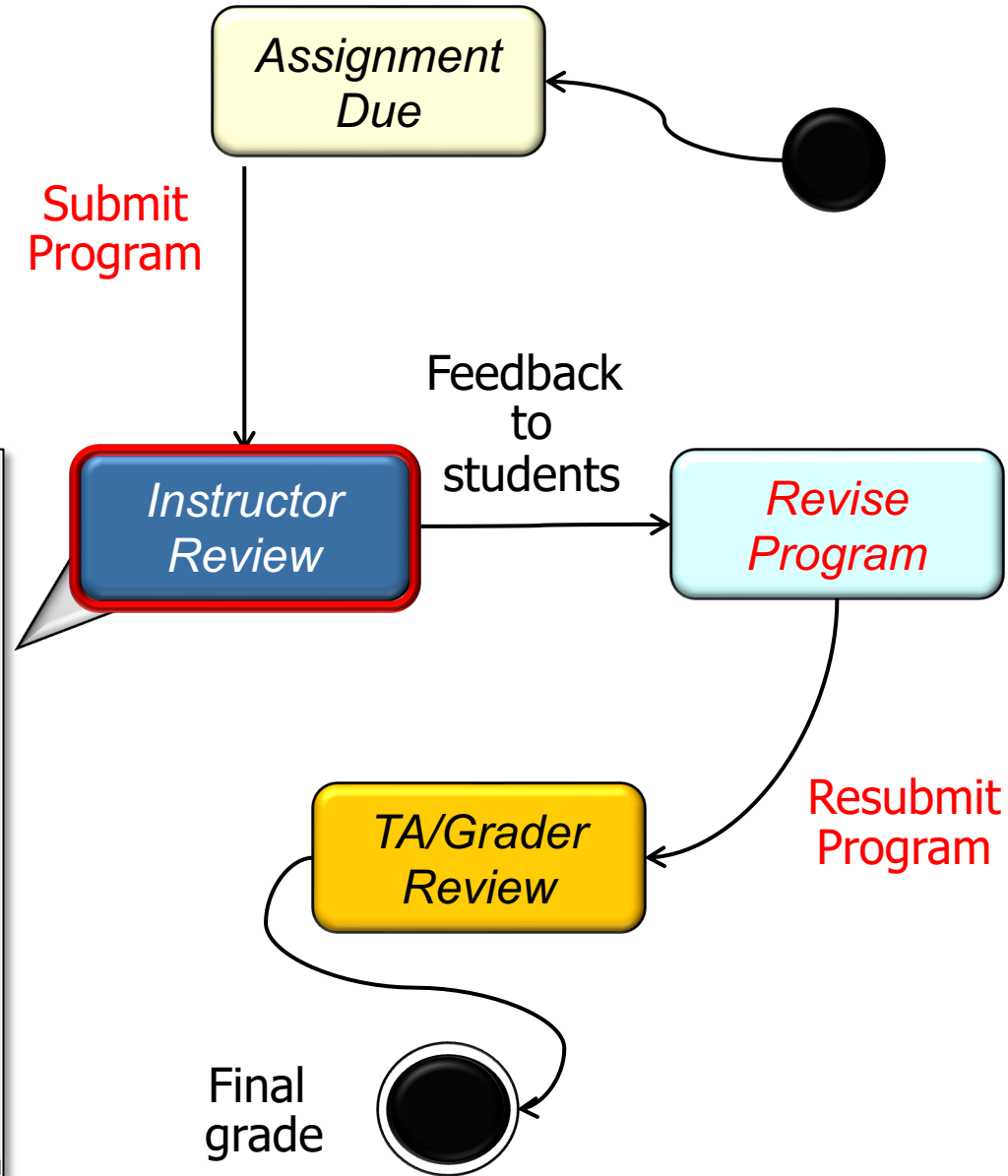
    // @@ This code is WAY too complicated. You just need to make
    // an instance of ProcessImageTask() and return it..
    Callable<Integer> callable = () ->
    {
        var imageUrl = urlConverter.apply(url);
        if(imageUrl == null) { return null; }
        var processImageTask = new ProcessImageTask(imageUrl);
        if(processImageTask == null) { return null; }
        return processImageTask.join();
    };
    return ForkJoinTask.adapt(callable);
}

protected ForkJoinTask<Integer> makeProcessImageTask(String url) {
    Function<String, URL> urlConverter = ExceptionUtils
    // @@ This is too complicated - just use URL::new:
    .rethrowFunction(urlStr -> new URL(urlStr));
    ...
}

public class URLCrawlerTask ...
    // @@ You don't need to have a 'count' field - instead, handle
    // this via local variables, which are thread-safe.
    private int count;

    URLCrawlerTask(String pageUri, int depth) {
        ...
        // @@ Don't allocate a new ForkJoinPool - you need to just
        // use the common fork-join pool.
        forkJoinPool = new ForkJoinPool();

        // @@ Don't do this, as per the comment above.
        count = 0;
        ...
    }
}
```

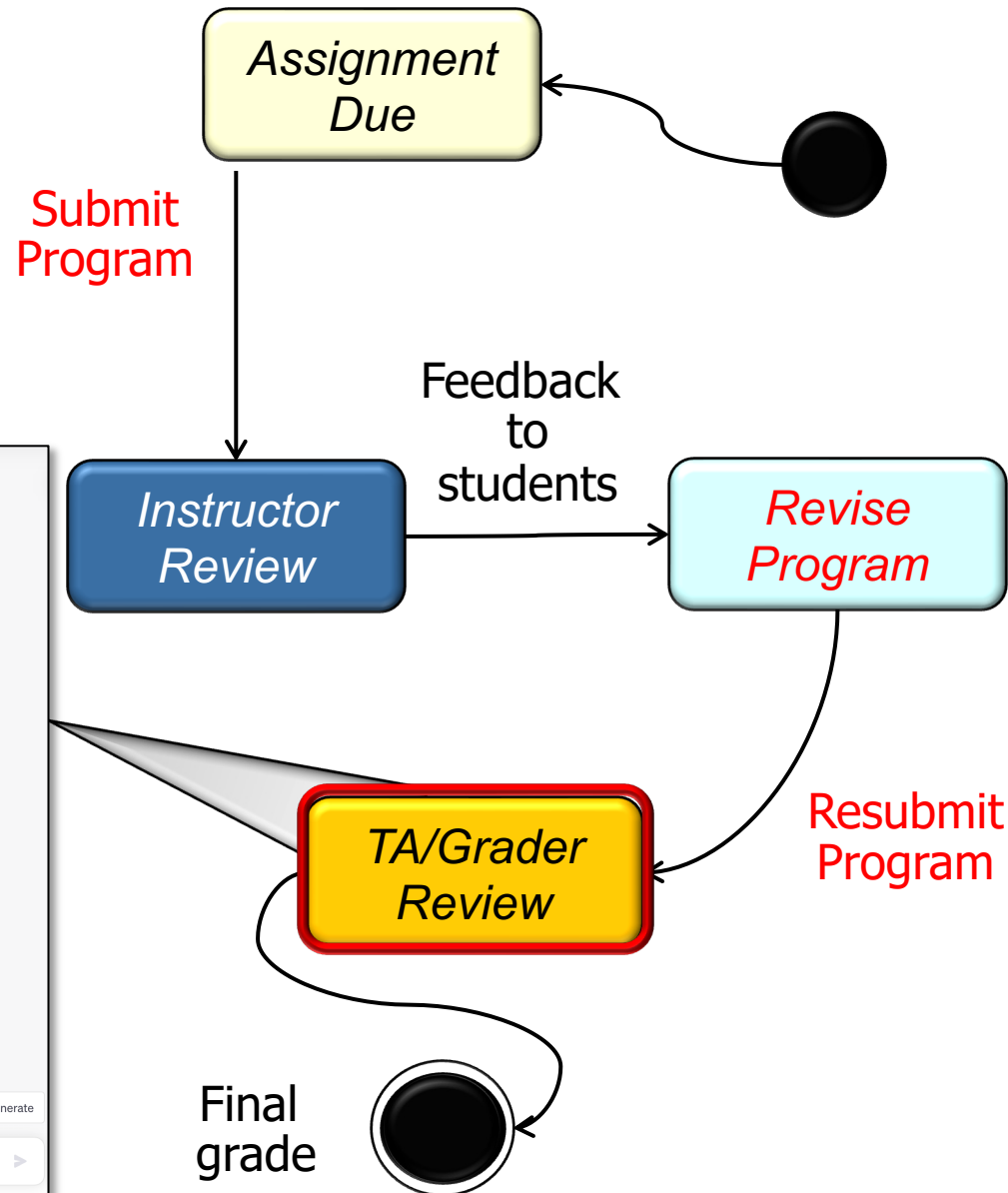




# Using AI+ to Review Programming Assignment Submissions

- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!
- Automating key steps in reviewing programming assignment solutions

```
Question 1:  
1. Carefully inspecting the `compute()` method, it has a call to `createNewCacheItem()` and `applyTransform()` but does not contain a direct call to the `submit()` method.  
2. Justification: The method only has calls to `createNewCacheItem()` and `applyTransform()`.  
  
No direct call to `submit()` method is present.  
  
Question 2:  
1. Inspecting the class, there are no direct references to the `mMaxDepth` field.  
2. Justification: `mMaxDepth` is not mentioned in any method or constructor.  
  
No methods directly access the `mMaxDepth` field.  
  
Question 3:  
1. Inspecting the class, no `try/catch` blocks are present in the provided code.  
2. Justification: No `try` or `catch` keywords are present.  
  
No `try/catch` blocks are used explicitly.  
  
Question 4:  
1. Inspecting the class, no usage of `AtomicReference` or `AtomicInteger` was found.  
2. Justification: No reference to `AtomicReference` or `AtomicInteger` exists.  
  
Neither `AtomicReference` nor `AtomicInteger` is used in the code.  
  
Question 5:  
1. Inspecting the class, no for-loops with explicit integer counter initialization, checking, and incrementing are present.  
2. Justification: There are no `for` loops.  
  
[Regenerate]  
[Send a message]
```



This step can benefit greatly from (semi)-automated means of assessing student solutions

# Using AI+ to Review Programming Assignment Submissions

- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!
- Automating key steps in reviewing programming assignment solutions
  - We've trained ChatGPT-4 to analyze student submissions for frequently made mistakes

## Question 1: Trace the `compute()` method

1. Inspecting the `compute()` method line-by-line.
2. No direct call to `submit()` method is observed.

## Question 2: Methods Accessing `mMaxDepth`

1. Inspecting each method line-by-line.
2. No methods directly access the `mMaxDepth` field.

## Question 3: Try/Catch Blocks

1. Inspecting each method line-by-line.
2. No try/catch blocks are explicitly used in the provided Java code.

## Question 4: Use of AtomicReference or AtomicInteger

1. Inspecting each method line-by-line.
2. No usage of AtomicReference or AtomicInteger is observed.

## Question 5: For Loops

1. Inspecting each method line-by-line.
2. No for loops meeting the specified criteria are found.

## Question 6: Direct Calls to `callInManagedBlocker()`

1. Inspecting each method line-by-line.
2. No methods directly call `callInManagedBlocker()`.

# Using AI+ to Review Programming Assignment Submissions

- AI+ helps improve reviewing of programming assignment submissions for my CS courses
  - Make specifications for my programming assignments that are harder to reverse-engineer!
- Automating key steps in reviewing programming assignment solutions
  - We've trained ChatGPT-4 to analyze student submissions for frequently made mistakes
    - LLMs are great for iterative improvements!



Your answer to question 6 was incorrect - there's a Java for-each loop in that method, but not a traditional index-based Java for loop. Please help me rephrase this question so you answer it correctly the first time.



Certainly, I understand that you'd like the question to focus specifically on the presence of traditional index-based Java 'for' loops as opposed to Java for-each loops in the `processPage()` method.

You could rephrase the question as follows:

"How many instances of traditional index-based 'for' loops are present in the 'processPage()' method of the 'URLCrawlerTask' class in the provided Java code? Please exclude Java for-each loops from this count."

This rephrased question makes it clear that the focus is on counting only traditional index-based 'for' loops, thereby excluding Java for-each loops.

Good luck doing this with your favorite static analysis tool or IDE!

# You are Part of History – So Enjoy the Time of Your Life!

Technological innovations over the past 1000 years that had the biggest impact on the world

1. Printing Press (15<sup>th</sup> Century)
2. Steam Engine (18<sup>th</sup> Century)
3. Electricity (19<sup>th</sup> Century)
4. Internet (20<sup>th</sup> Century)
5. Computer (20<sup>th</sup> Century)
6. Generative AI & AI+ (21<sup>st</sup> Century)



See [www.youtube.com/watch?v=uWbybukpdCU](https://www.youtube.com/watch?v=uWbybukpdCU)

---

End of the Applying AI+ to Review  
Programming Assignment  
Submissions in Vandy CS Courses