```java
public class SE421 {

	public static void main(String[] args) {
		print("Hello");

		/*
		 * TODO: print World in unicode
		 * \u002A\u002F\u0070\u0072\u0069\u006E\u0074\u0028\u0022\u0043\u0072\u0075\u0065\u006C\u0022\u0029\u003B\u002F\u002A
		 */
		print("World");
	}

	private static void print(String s){
		System.out.print(s + " ");
	}

}
```

Instructor: Ben Holland (ben-holland.com)

Learning Objectives

By the end of this course you should be able to:

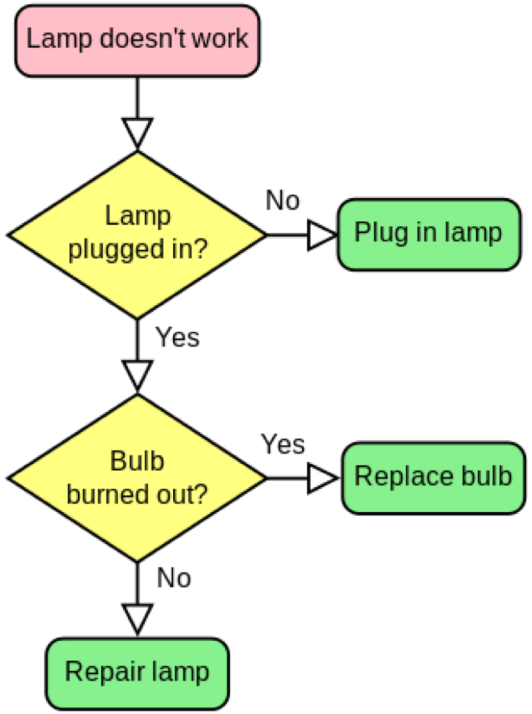
- Demonstrate basic bug hunting, exploitation, evasion, and post-exploitation skills
- Describe commonalities between vulnerability analysis and malware detection
- Describe fundamental limits in program analysis
- Challenge conventional viewpoints of security
- Confidently approach large third party software
- Critically evaluate software security products
- Locate additional relevant resources

Ethical Concerns

- Disclaimer: The content in this course was created for educational purposes only.
- Consider the consequences of your actions. *Remember that every action may have unforeseeable consequences.*

Ice Breaker Exercise: EIL5 “Programming”

- Explain It Like I’m Five (EIL5): How do computer programs work?
- Can your explanation intuitively address:
  - Complexity of software
  - Programming bugs
  - Security issues



Course Overview

- Course Website: https://se421.github.io
  - Review Syllabus!
  - Assignment 1 is available
- GitHub (course materials / assignments)
- Canvas (assignment submission / grades)
- Piazza (course help / discussions)

Do you agree?

- Antivirus protects us from modern malware.
- Antivirus protects us from yesterday’s threats.
- Antivirus protects us from last year’s threats.
- Antivirus is totally worthless.

Exercise (2014): Refactoring CVE-2012-4681

- “Allows remote attackers to execute arbitrary code via a crafted applet that bypasses SecurityManager restrictions…”
- CVE Created August 27th 2012 (~2 years old…)

Sample	Notes	Score (2014’s positive detections)
Original Sample	http://pastie.org/4594319	30/55
Technique A	Changed Class/Method names	28/55
Techniques A and B	Obfuscate strings	16/55
Techniques A-C	Change Control Flow	16/55
Techniques A-D	Reflective invocations (on sensitive APIs)	3/55
Techniques A-E	Simple XOR Packer	0/55