# The Java Programming Structure
## IFT 194: Lab 1

Brandon Doyle
bdoyle5@asu.edu

Dr. Usha Jagannathan
Usha.Jagannathan@asu.edu

July 1, 2018

# Part A

Content

# Part B

## 1 Poem

### 0.2 Poem

Content

# Conclusion

I spent approximately 5 hours completing this lab. The quickest portion was setting up my environment as I already had the JDK installed on my Linux machine and Eclipse.

Challenges I faced in writing this lab report were primarily around formatting. Because I've chosen LaTeXto present my code and findings,

```java
package lab_1;


public class Driver
{
    public static void main(String[] args)
    {
        System.out.println("Roses are red");
        System.out.println("Violets are blue");
        System.out.println("Sugar is sweet");
        System.out.println("And so are you!");
    }
}
```

Figure 1: Driver.java

```java
//********************************************************************************
// Count.java
// Brandon Doyle
// 1215232174
// Using comments in a Java program
// Demonstrate inline and single-line comment usage in Java programs.
//********************************************************************************

package lab_1;

public class Count
{
    /**
     * Print srings to the console.
     *
     * @param args Not used.
     */
    public static void main(String[] args)
    {
        // English
        System.out.println("one two three four five");

        // French
        System.out.println("un deux trois quatre cinq");

        // Spanish
        System.out.println("uno dos tres cuatro cinco");
    }
}
```

Figure 2: Count.java

```
#! /usr/bin/env python3.6
# -*- coding: utf-8 -*-
# vim:fenc=utf-8
#
# Copyright Â© 2018 Brandon Doyle <bjd2385@aperiodicity.com>
#
# Distributed under terms of the MIT license.

"""

"""

import cmd


class Visual(cmd.Cmd):
    """ simple cmd line example """

    prompt = 'Visual$ '

    def do_h(self, line: str) -> None:
        self.do_help(line)

    def do_greet(self, line) -> None:
        print('hello')

    def do_EOF(self, line) -> bool:
        return True

    def postloop(self) -> None:
        print()


if __name__ == '__main__':
    Visual().cmdloop()
```

Figure 3: test_onto.py