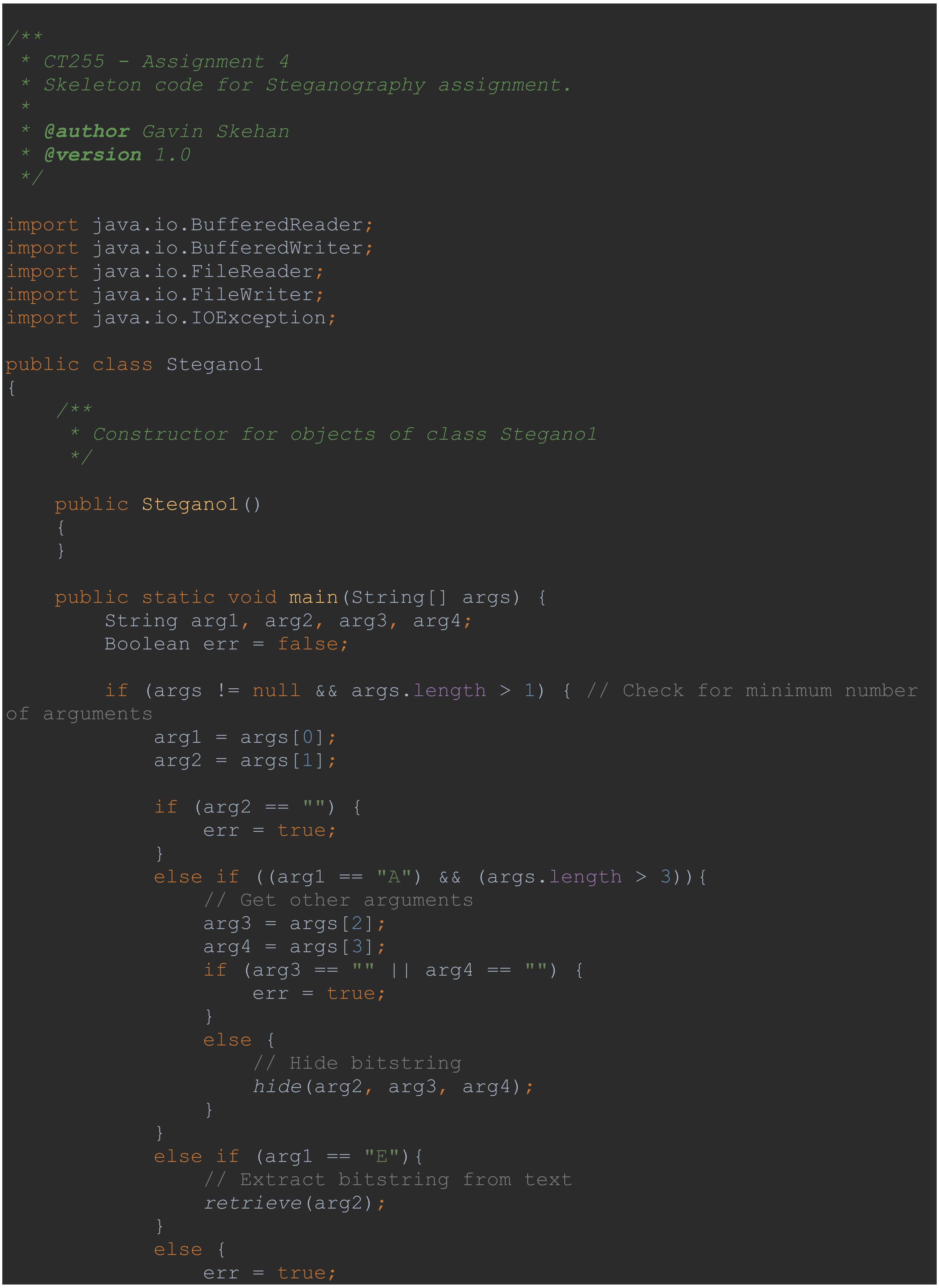
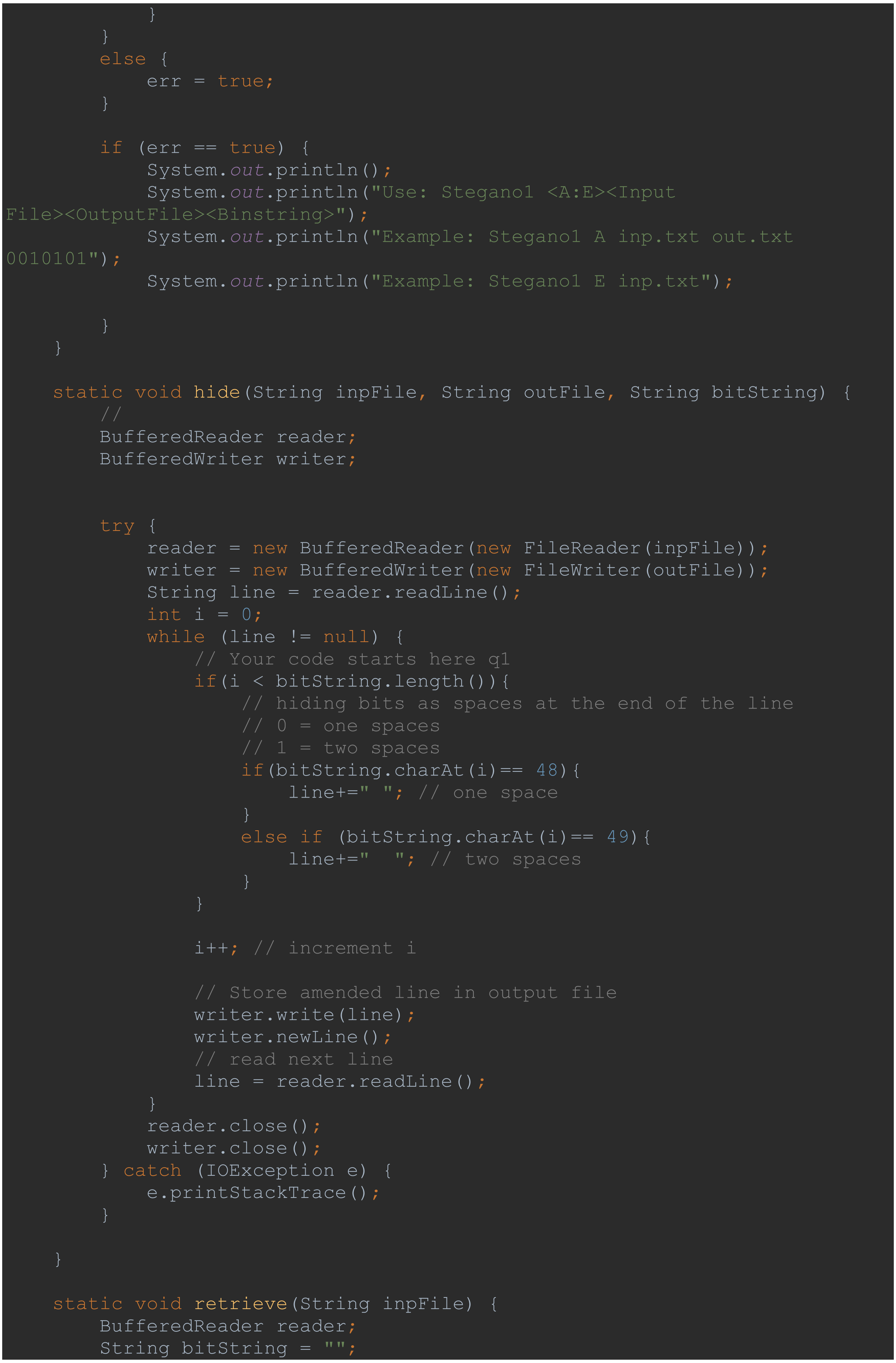
Next Generation Technologies Assignment 4 – Steganography

***I don't know how to run the code with the text file**

Problem 1 code:



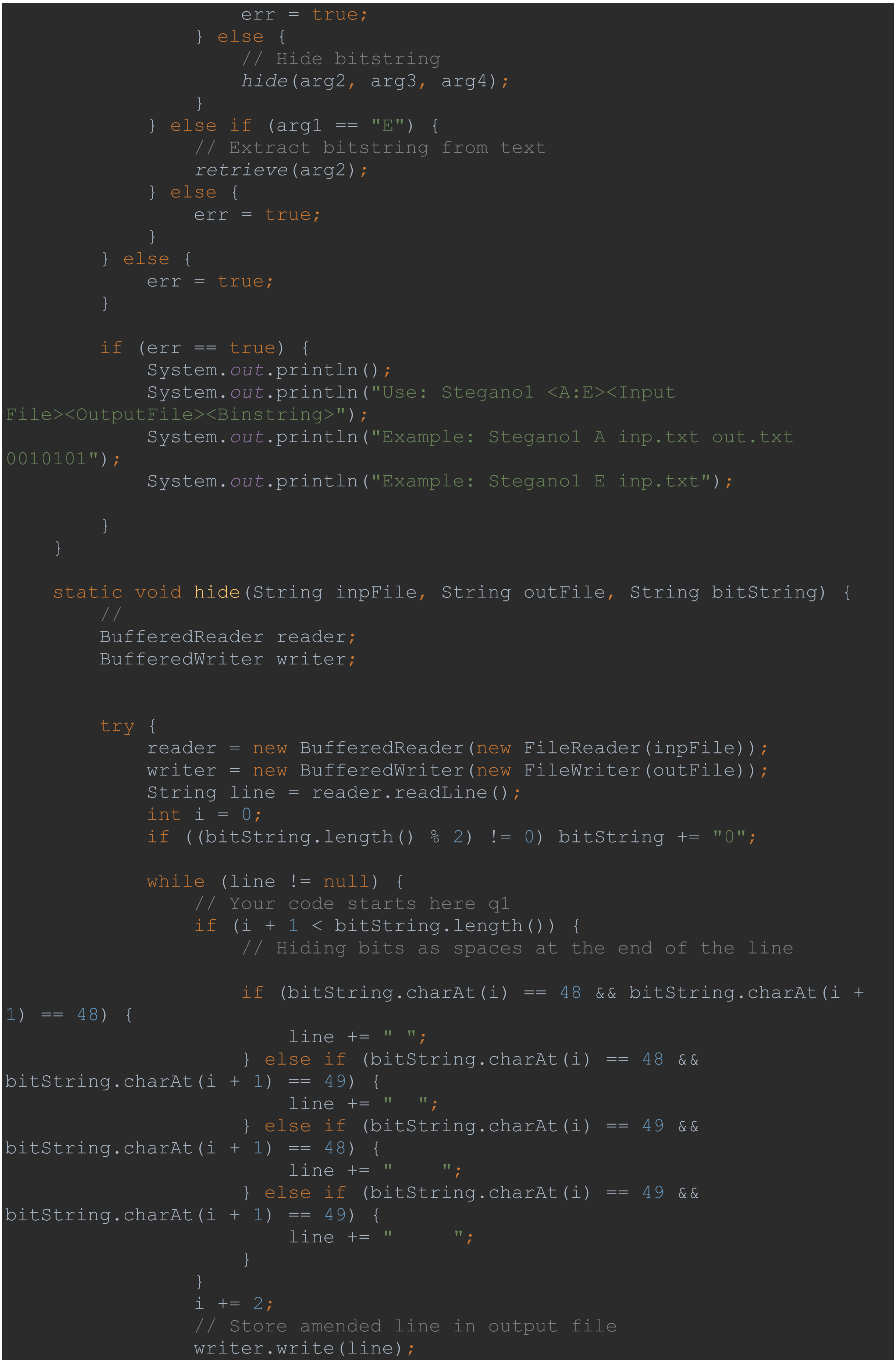
```java
/**
 * CT255 - Assignment 4
 * Skeleton code for Steganography assignment.
 *
 * @author Gavin Skehan
 * @version 1.0
 */

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Stegano1
{
    /**
     * Constructor for objects of class Stegano1
     */

    public Stegano1()
    {
    }

    public static void main(String[] args) {
        String arg1, arg2, arg3, arg4;
        Boolean err = false;

        if (args != null && args.length > 1) { // Check for minimum number of arguments
            arg1 = args[0];
            arg2 = args[1];

            if (arg2 == "") {
                err = true;
            }
            else if ((arg1 == "A") && (args.length > 3)){
                // Get other arguments
                arg3 = args[2];
                arg4 = args[3];
                if (arg3 == "" || arg4 == "") {
                    err = true;
                }
                else {
                    // Hide bitstring
                    hide(arg2, arg3, arg4);
                }
            }
            else if (arg1 == "E"){
                // Extract bitstring from text
                retrieve(arg2);
            }
            else {
                err = true;
```



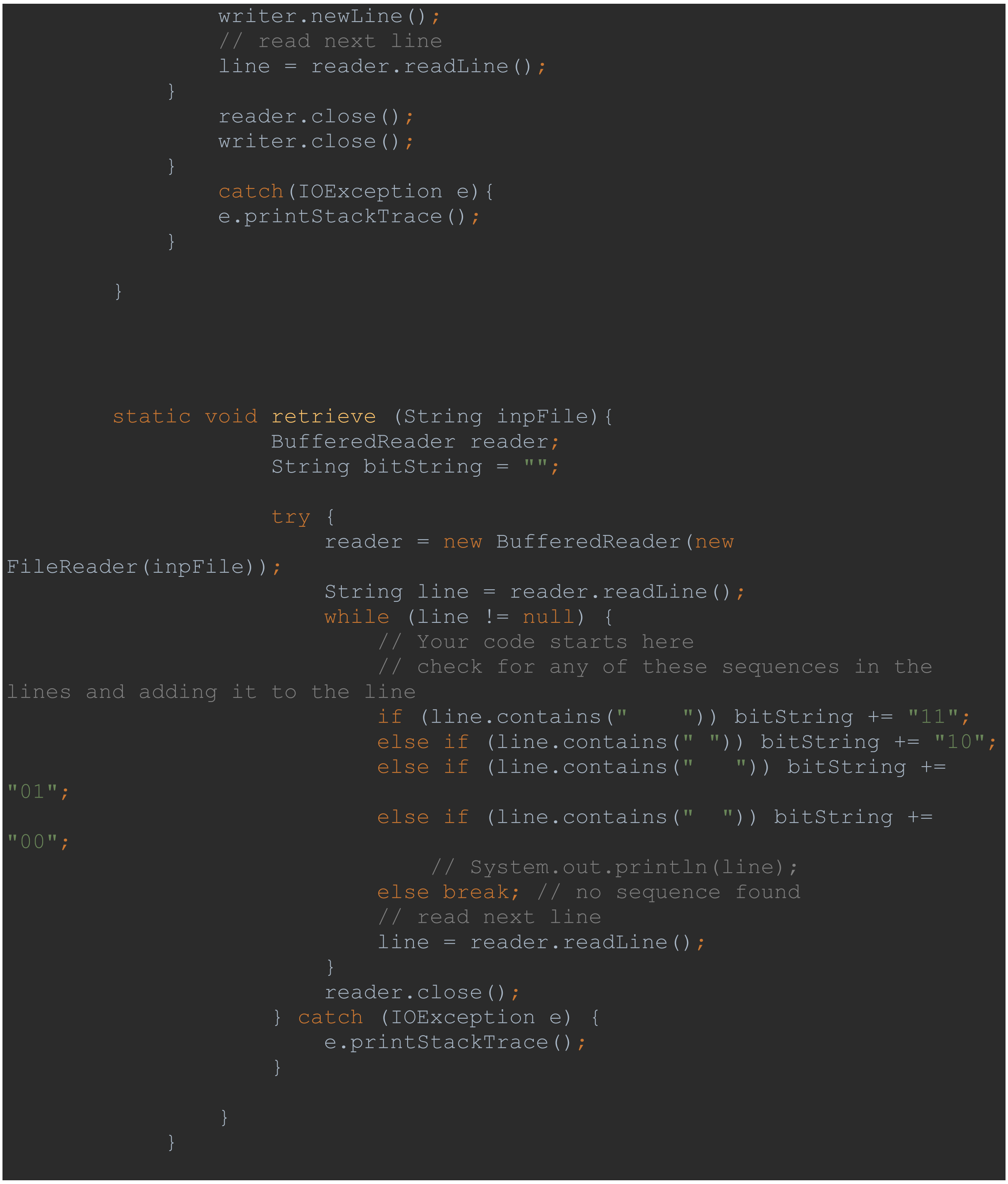
```
            }
        }
        else {
            err = true;
        }

        if (err == true) {
            System.out.println();
            System.out.println("Use: Stegano1 <A:E><Input
File><OutputFile><Binstring>");
            System.out.println("Example: Stegano1 A inp.txt out.txt
0010101");
            System.out.println("Example: Stegano1 E inp.txt");

        }
    }

    static void hide(String inpFile, String outFile, String bitString) {
        //
        BufferedReader reader;
        BufferedWriter writer;


        try {
            reader = new BufferedReader(new FileReader(inpFile));
            writer = new BufferedWriter(new FileWriter(outFile));
            String line = reader.readLine();
            int i = 0;
            while (line != null) {
                // Your code starts here q1
                if(i < bitString.length()){
                    // hiding bits as spaces at the end of the line
                    // 0 = one spaces
                    // 1 = two spaces
                    if(bitString.charAt(i)== 48){
                        line+=" "; // one space
                    }
                    else if (bitString.charAt(i)== 49){
                        line+="  "; // two spaces
                    }
                }

                i++; // increment i

                // Store amended line in output file
                writer.write(line);
                writer.newLine();
                // read next line
                line = reader.readLine();
            }
            reader.close();
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    static void retrieve(String inpFile) {
        BufferedReader reader;
        String bitString = "";
        try {
            reader = new BufferedReader(new FileReader(inpFile));
            String line = reader.readLine();
            while (line != null) {
                // Your code starts here
                if(line.contains("  "))bitString += "1"; // 2 spaces
                else if(line.contains(" "))bitString += "0"; // 1 space
                // System.out.println(line);
                else break;
                // read next line
                line = reader.readLine();
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
```

Problem 2 code:

```
/**
 * CT255 - Assignment 4
 * Skeleton code for Steganography assignment.
 *
 * @author Gavin Skehan
 * @version 1.0
 */

import java.io.*;

public class Stegano1 {
    /**
     * Constructor for objects of class Stegano1
     */

    public Stegano1() {
    }

    public static void main(String[] args) {
        String arg1, arg2, arg3, arg4;
        Boolean err = false;

        if (args != null && args.length > 1) { // Check for minimum number
of arguments
            arg1 = args[0];
            arg2 = args[1];

            if (arg2 == "") {
                err = true;
            } else if ((arg1 == "A") && (args.length > 3)) {
                // Get other arguments
                arg3 = args[2];
                arg4 = args[3];
                if (arg3 == "" || arg4 == "") {
```



```
                    err = true;
                } else {
                    // Hide bitstring
                    hide(arg2, arg3, arg4);
                }
            } else if (arg1 == "E") {
                // Extract bitstring from text
                retrieve(arg2);
            } else {
                err = true;
            }
        } else {
            err = true;
        }

        if (err == true) {
            System.out.println();
            System.out.println("Use: Stegano1 <A:E><Input
File><OutputFile><Binstring>");
            System.out.println("Example: Stegano1 A inp.txt out.txt
0010101");
            System.out.println("Example: Stegano1 E inp.txt");

        }
    }

    static void hide(String inpFile, String outFile, String bitString) {
        //
        BufferedReader reader;
        BufferedWriter writer;


        try {
            reader = new BufferedReader(new FileReader(inpFile));
            writer = new BufferedWriter(new FileWriter(outFile));
            String line = reader.readLine();
            int i = 0;
            if ((bitString.length() % 2) != 0) bitString += "0";

            while (line != null) {
                // Your code starts here q1
                if (i + 1 < bitString.length()) {
                    // Hiding bits as spaces at the end of the line

                    if (bitString.charAt(i) == 48 && bitString.charAt(i +
1) == 48) {
                        line += " ";
                    } else if (bitString.charAt(i) == 48 &&
bitString.charAt(i + 1) == 49) {
                        line += "  ";
                    } else if (bitString.charAt(i) == 49 &&
bitString.charAt(i + 1) == 48) {
                        line += "    ";
                    } else if (bitString.charAt(i) == 49 &&
bitString.charAt(i + 1) == 49) {
                        line += "     ";
                    }
                }
                i += 2;
                // Store amended line in output file
                writer.write(line);
```



```
                writer.newLine();
                // read next line
                line = reader.readLine();
            }
                reader.close();
                writer.close();
            }
                catch(IOException e){
                e.printStackTrace();
            }

        }




        static void retrieve (String inpFile){
                    BufferedReader reader;
                    String bitString = "";

                    try {
                        reader = new BufferedReader(new
FileReader(inpFile));
                        String line = reader.readLine();
                        while (line != null) {
                            // Your code starts here
                            // check for any of these sequences in the
lines and adding it to the line
                            if (line.contains("    ")) bitString += "11";
                            else if (line.contains(" ")) bitString += "10";
                            else if (line.contains("   ")) bitString +=
"01";

                            else if (line.contains("  ")) bitString +=
"00";
                                // System.out.println(line);
                            else break; // no sequence found
                            // read next line
                            line = reader.readLine();
                        }
                        reader.close();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }

                }
            }
```