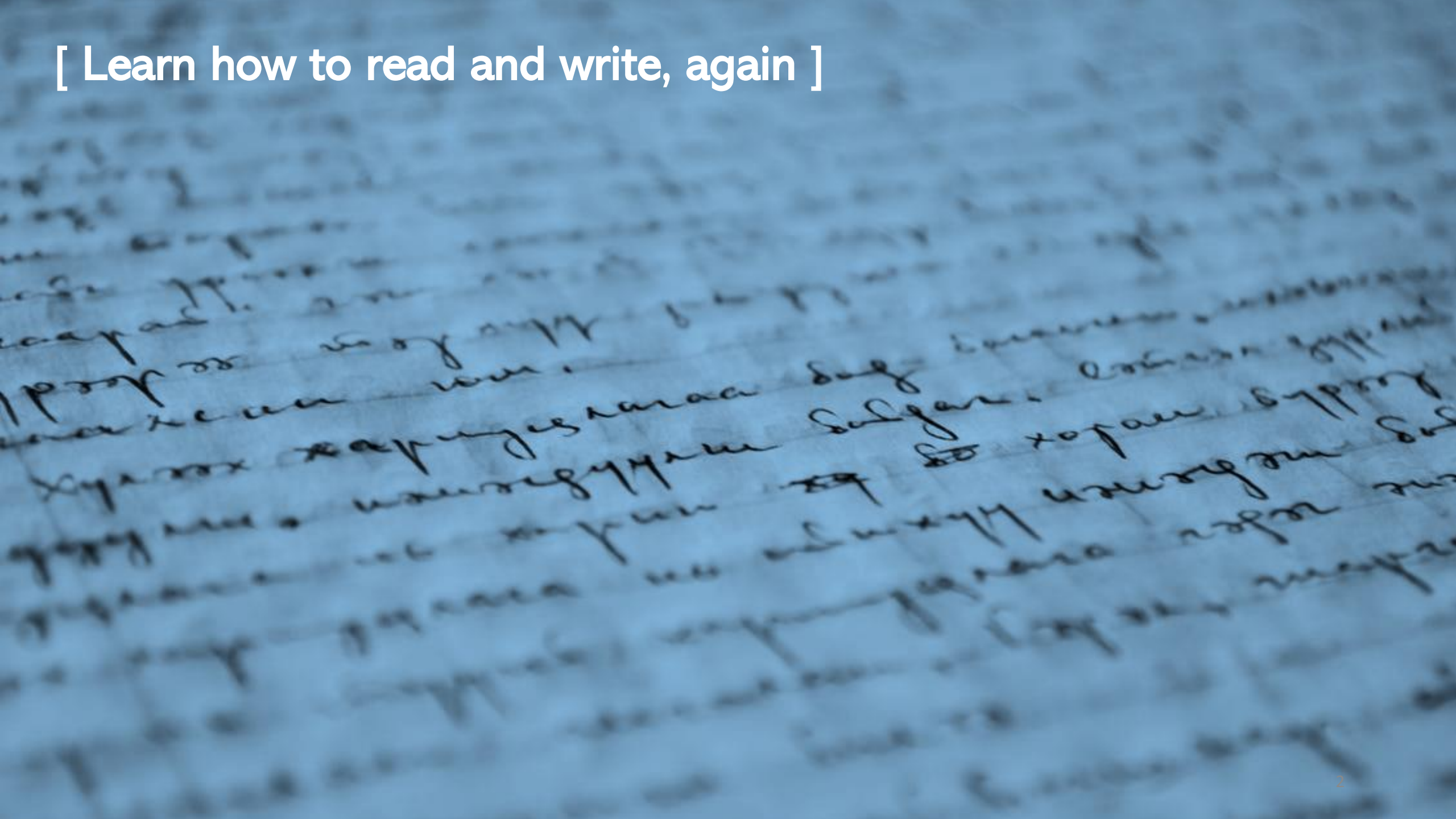
Foundations of Programming 2: Text I/O

FoP 2 Teaching Team, Faculty of Computer Science, Universitas Indonesia
Correspondence: Fariz Darari (fariz@cs.ui.ac.id)

UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER

[Learn how to read and write, again]



Why?

Computer programs are only useful if
they interact with the rest of the world in some way.

This interaction is referred to as Input/Output (or I/O).

We have learned how to write to console

```
System.out.println("Hey, Tayo!");
```

How to read from console

```
// inside main

System.out.println("Please type your name and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	String name = reader.readLine();
	System.out.println("Hello, " + name + "!");
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

How to read from console

```
// inside main

System.out.println("Please type your name and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	String name = reader.readLine();
	System.out.println("Hello, " + name + "!");
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

In Java, console input is accomplished by reading from **`System.in`**

How to read from console

```
// inside main

System.out.println("Please type your name and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	String name = reader.readLine();
	System.out.println("Hello, " + name + "!");
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

In Java, console input is accomplished by reading from **`System.in`**
`InputStreamReader` converts bytes to characters

How to read from console

```
// inside main

System.out.println("Please type your name and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	String name = reader.readLine();
	System.out.println("Hello, " + name + "!");
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

In Java, console input is accomplished by reading from **`System.in`**
`InputStreamReader` converts bytes to characters
`BufferedReader` optimizes the reading by adding buffering feature (like YouTube)

Quiz time: Count the number of words of user input in console

Quiz time: Count the number of words of user input in console

```
// inside main

System.out.println("Please type something and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	System.out.println(reader.readLine().split(" ").length);
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

Quiz time: Count the number of words of user input in console (better)

```
// inside main

System.out.println("Please type something and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	System.out.println(reader.readLine().split("\\s+").length);
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

Quiz time: Read first 5 characters of console input (Hint: read() method of BufferedReader could be helpful)

Quiz time: Read first 5 characters of console input (Hint: read() method of BufferedReader could be helpful)

```
System.out.println("Please type something and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	int i = 0;
	while(i < 5) {
		System.out.print((char) reader.read());
		i++;
	}
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

Quiz time: Read first 5 characters of console input (Hint: read() method of BufferedReader could be helpful)

```
System.out.println("Please type something and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	int i = 0;
	while(i < 5) {
		System.out.print((char) reader.read());
		i++;
	}
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

However, this might be problematic if you enter Enter (= line break).

Quiz time: Read first 5 characters of console input

```
System.out.println("Please type something and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	System.out.println(reader.readLine().substring(0, 5));
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

Quiz time: Read first 5 characters of console input

```
System.out.println("Please type something and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	System.out.println(reader.readLine().substring(0, 5));
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

What could be a problem for this "solution"?

Quiz time: Read first 5 characters of console input

```
System.out.println("Please type something and press Enter.");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
	String line = reader.readLine();
	System.out.println(line.substring(0, line.length() < 5? line.length(): 5));
} catch(IOException e) {
	System.out.println("An error occurred: " + e.getMessage());
}
```

How to read from text files

```
public static List<String> getAllLines(String filename) throws IOException {
	List<String> lines = new ArrayList<String>();
	try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
		String line = br.readLine();
		while (line != null) {
			lines.add(line);
			line = br.readLine();
		}
	}
	return lines;
}
```

How to read from text files

```
public static List<String> getAllLines(String filename) throws IOException {
	List<String> lines = new ArrayList<String>();
	try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
		String line = null;
		while ((line = br.readLine()) != null) {
			lines.add(line);
		}
	}
	return lines;
}
```

How to read from text files

```
durian
mango
mangosteen
apple
```

fruit.txt

```
// inside main

try {
	System.out.println(getAllLines("data/fruit.txt"));
} catch (IOException e) {
	e.printStackTrace();
}
```

Quiz: Only read lines that contain "o"

Quiz: Only read lines that contain "o"

```
public static List<String> getAllLinesWitho(String filename) throws IOException {
	List<String> lines = new ArrayList<String>();
	try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
		String line = null;
		while ((line = br.readLine()) != null) {
			if(line.contains("o"))
				lines.add(line);
		}
	}
	return lines;
}
```

Quiz: Create SearchAndPrint.java that:

- takes two arguments: textToSearch and filename
- print every line where textToSearch occurs
- add a summary at the end of the printing, showing the total number of lines containing textToSearch

Quiz: Create SearchAndPrint.java that:

- takes two arguments: textToSearch and filename
- print every line where textToSearch occurs
- add a summary at the end of the printing, showing the total number of lines containing textToSearch

Solution:
http://ocw.ui.ac.id/pluginfile.php/1479/mod_resource/content/0/SearchAndPrint.java

File class, what's the output?

```
File f = new File("data/someFile.txt");
System.out.println(f.exists());
System.out.println(f.getName());
System.out.println(f.getAbsolutePath());
System.out.println(f.getParent());
System.out.println(f.isDirectory());
System.out.println(f.isFile());
System.out.println(f.delete());
System.out.println(f.exists());
```

PrintWriter class

```
try {
	File f = new File("data/someFile.txt");
	if(f.exists()) {
		System.out.println("File already exists!");
	} else {
		PrintWriter pw = new PrintWriter(f);
		pw.println("Budi");
		pw.print("Ani");
		pw.println("Candra\n");
		pw.println("Dedi");
		pw.close();
	}
} catch(IOException e) {
	e.printStackTrace();
}
```

PrintWriter class with try-with

```
try {
	File f = new File("data/someFile.txt");
	if(f.exists()) {
		System.out.println("File already exists!");
	} else {
		try(PrintWriter pw = new PrintWriter(f);) {
			pw.println("Budi");
			pw.print("Ani");
			pw.println("Candra\n");
			pw.println("Dedi");
		}
	}
} catch(IOException e) {
	e.printStackTrace();
}
```

PrintWriter class with try-with

```
try {
	File f = new File("data/someFile.txt");
//	if(f.exists()) {
//		System.out.println("File already exists!");
//	} else {
		try(PrintWriter pw = new PrintWriter(f);) {
			pw.println("Budiarti");
			pw.print("Aniarto");
			pw.println("Candraka\n");
			pw.println("Dedikasi");
		}
//	}
} catch(IOException e) {
	e.printStackTrace();
}
```

What happens if a file with that name already exists?

PrintWriter class with try-with

```
try {
	File f = new File("data/someFile.txt");
//	if(f.exists()) {
//		System.out.println("File already exists!");
//	} else {
		try(PrintWriter pw = new PrintWriter(new FileWriter(f,true));) {
			pw.println("Budiarti");
			pw.print("Aniarto");
			pw.println("Candraka\n");
			pw.println("Dedikasi");
		}
//	}
} catch(IOException e) {
	e.printStackTrace();
}
```

What happens if a file with that name already exists?

Quiz time: Replace every line with the word "SECRET" to "Lorem ipsum dolor" in a text file

Quiz time: Replace every line with the word "SECRET" to "Lorem ipsum dolor" in a text file

```
public static void main(String[] args) {

	// inside main
	String fileLoc = "data/someFile.txt";
	File f = new File(fileLoc);
	if(f.exists()) {
		List<String> strList = null;
		try {
			strList = getAllLines(fileLoc);
			f.delete();
			try(PrintWriter pw = new PrintWriter(fileLoc);) {
				for(String str:strList) {
					if(str.contains("SECRET"))
						pw.println("Lorem ipsum dolor");
					else
						pw.println(str);
				}
			}
		} catch (IOException e) {
			e.printStackTrace();
		}
	}

}
```


https://gist.github.com/fadirra/b87f8510cdddce73540f980794d13a5e

Bonus: Lorem ipsum meaning

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but occasionally circumstances occur in which toil and pain can procure him some great pleasure.

Inspired by:

https://en.wikibooks.org/wiki/Java_Programming
Liang. Introduction to Java Programming. Tenth Edition. Pearson 2015.
Think Java book by Allen Downey and Chris Mayfield.
Eck. Introduction to Programming Using Java. 2014.