

# Long Method

[Blazej.Pietrzak@cs.put.poznan.pl](mailto:Blazej.Pietrzak@cs.put.poznan.pl)

# What is Long Method?

- A method that is too long and therefore cannot be entirely located on the screen
  - Assume  $LOC > 20$  on laboratory classes
- Cramped methods  
(i.e. complicated expressions)
- Complex methods
  - Assume  $CC > 4$  on laboratory classes

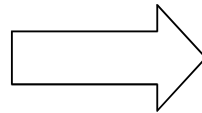
# Long Method – Motivation

- Programs that live best and longest are those with short methods.
- All the payoffs of indirection – explanation, sharing and choosing – are supported by little methods.

# Long Method -examples

Method is too long (e.g. LOC > 20)

```
public void init() {  
    //font init  
    font1.init();  
    ...  
    font6.init();  
  
    //logs init  
    logger1.init();  
    ...  
    logger6.init();  
}
```



```
public void init() {  
    initFonts();  
    initLogs();  
}  
public void initFonts(){  
    font1.init();  
    ...  
    font6.init();  
}  
public void initLogs() {  
    logger1.init();  
    ...  
    logger6.init();  
}
```

# Long Method -examples cont.

Too complex method (cyclomatic complexity > 4)

```
public void read(String path) {  
    try {  
        Reader in = new FileReader(path);  
  
        for (int ch = in.read(); ch != -1; ch = in.read()) {  
            if (ch == 'a') continue;  
            if (ch == 'b') { ... }  
        }  
  
        in.close();  
    } catch (IOException ex) {  
        ...  
    } catch (FileNotFoundException ex) { ... }  
}
```

Cyclomatic Complexity = 6

# Long Method -examples cont.

OK

```
public void read(String path) {  
    try {  
        read(new FileReader(path));  
    } catch (IOException ex) {  
        ...  
    } catch (FileNotFoundException ex) { ... }  
}
```

Cyclomatic Complexity = 3

```
public void read(Reader in) throws IOException {  
    for (int ch = in.read(), ch != -1; ch = in.read()) {  
        if (ch == 'a') continue;  
        if (ch == 'b') { ... }  
    }  
}
```

Cyclomatic Complexity = 4