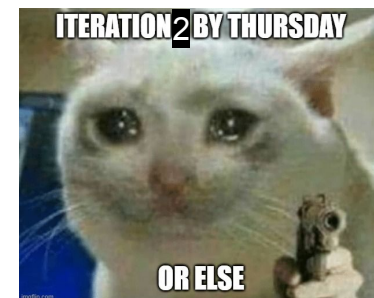


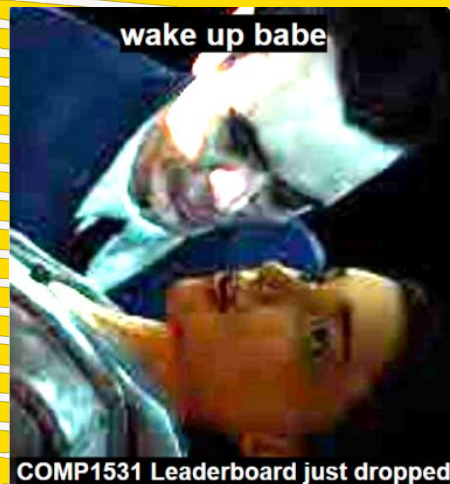
COMP1531 | T09B / H17B

# Week 7

[william.huynh3@unsw.edu.au](mailto:william.huynh3@unsw.edu.au)



Lab05 Deploy be like



UNSW  
SYDNEY

# Learning objectives

- DRY-KISS

Dry Kiss

# DryKiss: Writing maintainable code

---

DryKiss is a programming principle that stands for:

Don't repeat yourself, Keep it Simple Stupid

It aims to reduce repetition by leveraging functions combined with a sprinkle of common sense



DryKiss

DryKiss Principles are worth 30% of Iteration 3



```

1  ✓ const datastore = {
2      users: [], // array of user objects
3      channels: [], // array of channel objects
4      tokens: [] // array of token objects
5  }
6
7  ✓ function join_channel(token, channelId) {
8      authUserId = -1; // Initially invalid
9  ✓  for (const storedToken of datastore.tokens) {
10  ✓      if (storedToken.token == token) {
11          | authUserId = storedToken.token;
12          | }
13      }
14  ✓  if (authUserId == -1) {
15      | return {error: "error"};
16      | }
17      channelIdSearch = -1; // Initially invalid
18  ✓  for (const channel of datastore.channels) {
19  ✓      if (channel.id == channelId) {
20          | channelIdSearch = channel.id;
21          | }
22      }
23  ✓  if (channelIdSearch == -1) {
24      | return {error: "error"};
25      | }
26
27  ✓  for (const channel of datastore.channels) {
28  ✓      if (channel.id == channelId) {
29          | channel.members.push(authUserId);
30          | }
31      }
32      return {};
33  }

```

What's wrong with this code?

There's a lot of **repetition**!

It is **not simple code!!!!**

```
20  function join_channel(token, channelId) {
21      authUserId = -1; // Initially invalid
22      for (const storedToken of datastore.tokens) {
23          if (storedToken.token == token) {
24              authUserId = storedToken.token;
25          }
26      }
27      if (authUserId == -1) {
28          return {error: "error"};
29      }
30      channelIdSearch = -1; // Initially invalid
31      for (const channel of datastore.channels) {
32          if (channel.id == channelId) {
33              channelIdSearch = channel.id;
34          }
35      }
36      if (channelIdSearch == -1) {
37          return {error: "error"};
38      }
39      for (const channel of datastore.channels) {
40          if (channel.id == channelId) {
41              channel.members.push(authUserId);
42          }
43      }
44      return {};
45  }
46 }
```

What's wrong with this code?

```
1  const datastore = {
2    users: [], // array of user objects
3    channels: [], // array of channel objects
4    tokens: [] // array of token objects
5  }
6
7  function post_message(token, message) {
8    authUserId = get_userId_from_token(token);
9    if (!authUserId) {
10     return {error: "error"};
11   }
12   if (message.length() > 1000) {
13     return {error: "error"};
14   }
15   /*
16    Do the rest of the stuff...
17   */
18
19 }
20
21 function edit_message(token, message, messageId) {
22   authUserId = get_userId_from_token(token);
23   if (!authUserId) {
24     return {error: "error"};
25   }
26   if (message.length() > 1000) {
27     return {error: "error"};
28   }
29   /*
30    Do the rest of the stuff...
31   */
32 }
```



```

1  const datastore = {
2      users: [], // array of user objects
3      channels: [], // array of channel objects
4      tokens: [] // array of token objects
5  }
6
7  function post_message(token, message) {
8      authUserId = get_userId_from_token(token);
9      if (!authUserId) {
10         return {error: "error"};
11     }
12     if (message.length() > 1000) {
13         return {error: "error"};
14     }
15     /*
16      Do the rest of the stuff...
17     */
18
19 }

```

```

20
21 function edit_message(token, message, messageId) {
22     authUserId = get_userId_from_token(token);
23     if (!authUserId) {
24         return {error: "error"};
25     }
26     if (message.length() > 1000) {
27         return {error: "error"};
28     }
29     /*
30      Do the rest of the stuff...
31     */
32 }

```

It is not **maintainable**!



- Use `const` instead of `let`
- Use `for-of` instead of `for-index`
- Use `npm packages` where possible
- Use `in-built JS functions` where possible
  - `.includes()`
  - `.map()`
  - `.sort()`
- Other things from COMP1511
  - Good variable names
  - Appropriate indentation
  - Come back to this idiot

### Following DryKiss Principles

- Making use of `functions` wherever possible
- Making code maintainable
  - Storing functions in `appropriately named files`
  - Regular, appropriate `comments`
  - `JSDoc`

```

1 import promptSync from 'prompt-sync';
2
3 /**
4  * Given an array of n integers, calculate the minimum, maximum, and the
5  * product of the first n-1 numbers and last n-1 numbers.
6  */
7 function drykiss(myList: number[]) {
8   let myMin = Infinity;
9   for (let i = 0; i < myList.length; i++) {
10    if (myList[i] < myMin) {
11     myMin = myList[i];
12    }
13   }
14
15   let myMax = -Infinity;
16   for (let i = 0; i < myList.length; i++) {
17    if (myList[i] > myMax) {
18     myMax = myList[i];
19    }
20   }
21
22   let prod = 1;
23   for (let i = 0; i < 4; i++) {
24    prod = prod * myList[i];
25   }
26   const prodHead = prod;
27
28   prod = 1;
29   for (let i = 1; i < 5; i++) {
30    prod = prod * myList[i];
31   }
32
33   const result = [myMin, myMax, prodHead, prod];
34   return result;
35 }
36

```

# Activity

One of your teammates has written their functions for the project and made a **merge request** into master for their branch.

However, when reviewing their merge request, you realise that their code is of **extremely poor style** and **does not adhere to Dry-Kiss** principles at all.

Since the team-mate has “work” and cannot fix it themselves, **it is up to you and the rest of your team to fix it for them!**

Task: Rewrite the fool’s code to adhere to [DryKiss](#)