Welcome to CS2030 Lab 1!

28 January 2021 [10J]

`whoami`

Chen **YiJia**

- Year 3 Mathematics
- Interests: Violin, Anime, ...
- Second time teaching CS2030
- is currently behind on lectures 🥲

who are you?

1. name
2. year + major
3. programming languages
4. fun fact/hobbies/flex

What does CS2030/S teach you?

Java?
Vim?
Linux?
Terminal?

What does CS2030/S teach you?

~~Java?~~
~~Vim?~~
~~Linux?~~
~~Terminal?~~

means you must self-study
🙂
you can request from me if
you need help

Programming Paradigms
Design Patterns

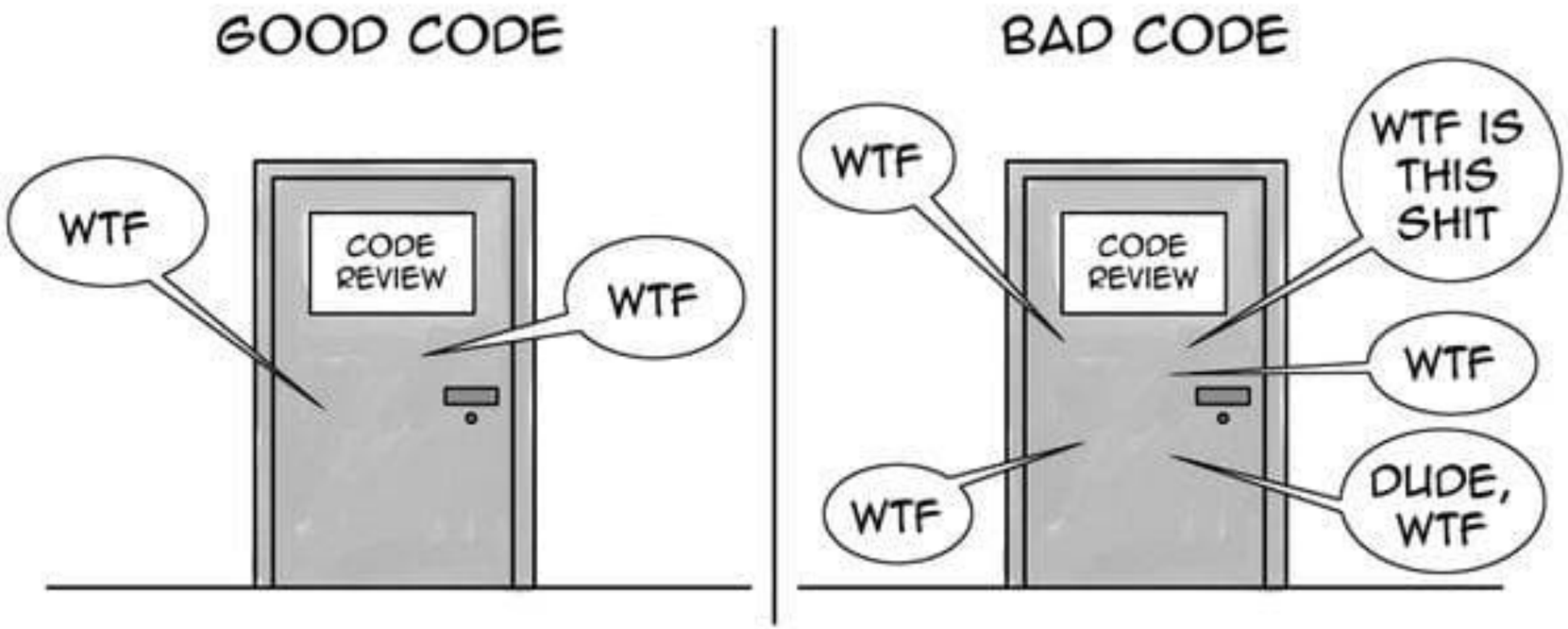
more practice, less theory

What does CS2030/S teach you?

Programs are meant to be read by humans and only incidentally for computers to execute.

— *Harold Abelson and Gerald Jay Sussman*

What does CS2030/S teach you?



Admin Information

Continuous Assessments	**Weightage (60%)**
Practical Assessment 1	15
Practical Assessment 2	20
Individual Project	10
Weekly Labs	10
Class Participation and Peer Learning Activities (github issues/wiki)	5


PLEASE STAND BY

Java Primer: String equality

- Use `.equals()` instead of `==` to compare Strings.
- `==` compares memory addresses.

Java Primer: Class equality

- Java classes have their own implementation of `.equals()`. Read the API).
- When creating your own custom class, `.equals()` is *inherited* from the `Object` class.
- Default implementation of `.equals()` is to compare memory addresses, so calling `.equals()` within your custom class is the same as using ==.
- *Override* the `.equals()` method with an implementation that you want to use.

Java Primer: toString

- JShell uses a class's `toString` implementation to print output.
- Override the `toString` method to make it print a custom string.

Java Primer: `@Override` tag

```
@Override
public String toString() {
    return "Hello World";
}
```

- Not necessary to include it, but good to have.
- If the tag is included, the code will not compile if the method is not overriden properly.
- Useful for checking that you have overriden the correct method/overriden the method properly.

Java Primer: Passing by value

- In Java, variables are passed by value.
- A copy of the information that is stored in the variable is passed into methods as parameters.

Java Primer: Passing by value

```
private static void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}


int a = 1;
int b = 2;
swap(a, b);
System.out.println("a: " + a + ", b: " + b);

[Output] a: 1, b: 2 # Swap does not occur
```

- Primitive types store the actual values.
- A copy of the value is passed to the swap method.

Java Primer: Passing by value

```
private static void swap(int[] array, int a, int b) {
    int temp = array[a];
    array[a] = array[b];
    array[b] = temp;
}
```

```
int[] array = {0, 1};
swap(array, 0, 1);
System.out.println("first entry: " + array[0] + ", second entry: " + array[1]);
```

```
[Output] first entry: 1, second entry: 0 # Swap occurs
```

- Reference types (classes) store memory addresses.
- A copy of the memory address is passed to the `swap` method.

Java Style: Avoid magic numbers

```
int numberOfMinutes = numberOfSeconds / 60;
```

What does 60 mean in the above line of code? You may have guessed that 60 represents the number of seconds in one minute. However, you only knew because you had the benefit of prior contextual knowledge (i.e. you know about the concept of time, and could thus infer what the number 60 meant).
We refer to 60 as a **magic number**; you need prior context to understand it. In general, we try to avoid magic numbers to make our code is more readable.

Java Style: Avoid magic numbers

```
private static final int NUMBER_OF_SECONDS_IN_ONE_MINUTE = 60;
int numberOfHours = numberOfSeconds / NUMBER_OF_SECONDS_IN_ONE_MINUTE;
```

- We give magic numbers meaning by assigning them to constants.
- Variables which hold constant values typically have `static` and `final` keywords.
- Their names are also canonically written in all uppercase with consecutive words being separated by underscores.
- Benefit: You only need to change the value once if you use it in multiple places.

Java Primer: Comparing floating point numbers

```
if (double 1 == double 2) {
    // do something
}
```

- You may be accustomed to doing something like the above to compare to floating point numbers (i.e. numbers with some value after the decimal point, like 1.0, 123.123, etc.).
- The above code does not always work!
- It has to do with how floating point numbers are represented in Java (you will learn more if you take CS2100).

Java Primer: Comparing floating point numbers

```
private static final double THRESHOLD = 1E-15; // 10^-15
if (Math.abs(double1 - double2) ⩽ THRESHOLD) {
    // do something
}
```

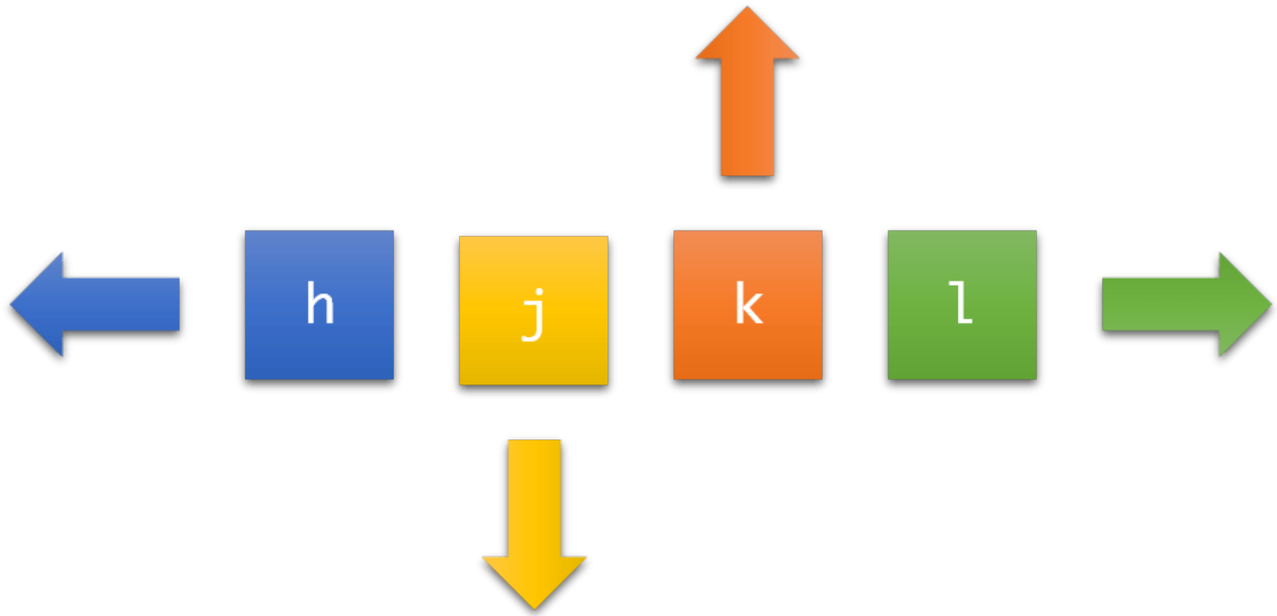
- We need to do something like the above instead.
- Basically, we check that the difference between both numbers is smaller than some small threshold value when comparing for “equality”.
- `Math.abs` takes the absolute (non-negative) value of the number being passed to it.

Common Bash (Linux) Commands

Action	**Usage**
List directory contents	`ls`
Renaming a file	`mv` oldname newname
Deleting a file	`rm` filename
Auto-complete	Tab ⇆
Navigating between old commands	⬆️⬇️

Common Vim Commands

Action	Usage
Indent all code in file	`gg=G` (`gg` moves the cursor to the beginning of the file)
Undo	`U`
Auto-complete	`Ctrl + N`
Jump to start of line	`0`
Jump to end of a line	`$`
Delete a line	`DD`

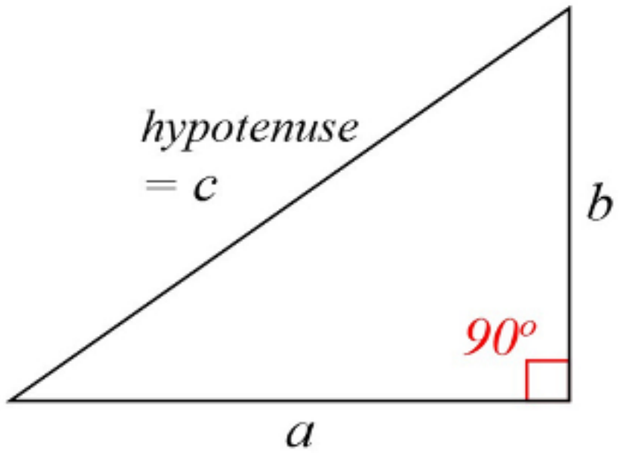


Common Vim Commands

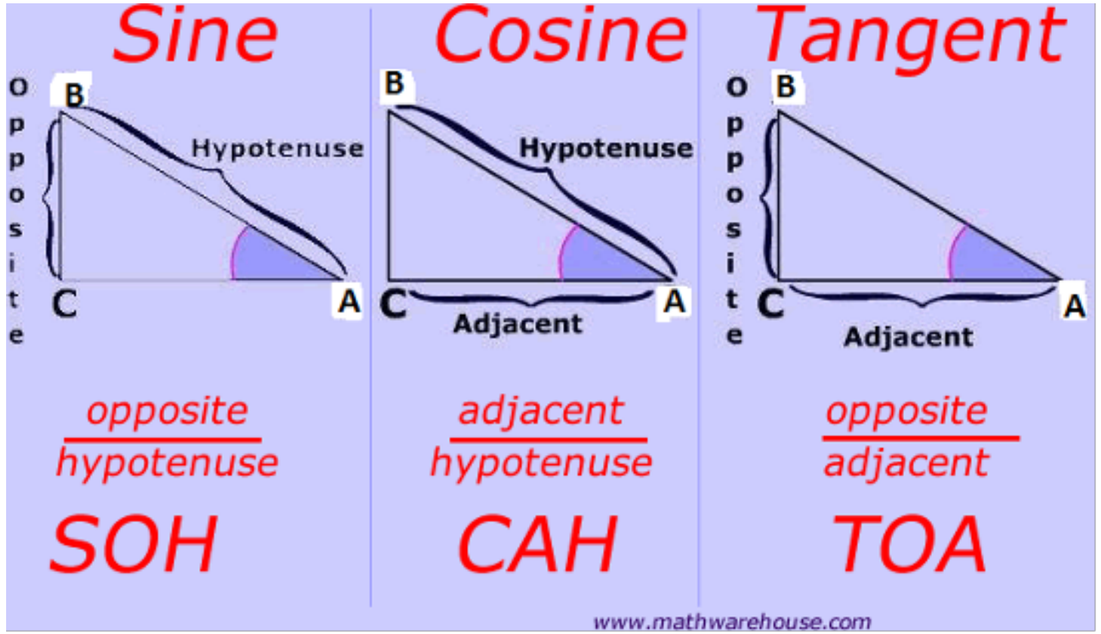
Action	Usage
Enter Visual-Line mode	`Shift + V`
[Visual Line] Move cursor	➡️⬅️⬆️⬇️
[Visual Line] Delete	`D`
[Visual Line] Cut	`X`
[Visual Line] Copy/Yank	`Y`
[Visual Line] Paste	`P`
Split window horizontally	`:sp`
Split window vertically	`:vsp`
Navigate between split windows	`Ctrl + W`
New tab	`:tabnew`

Lab 1: Math Recap

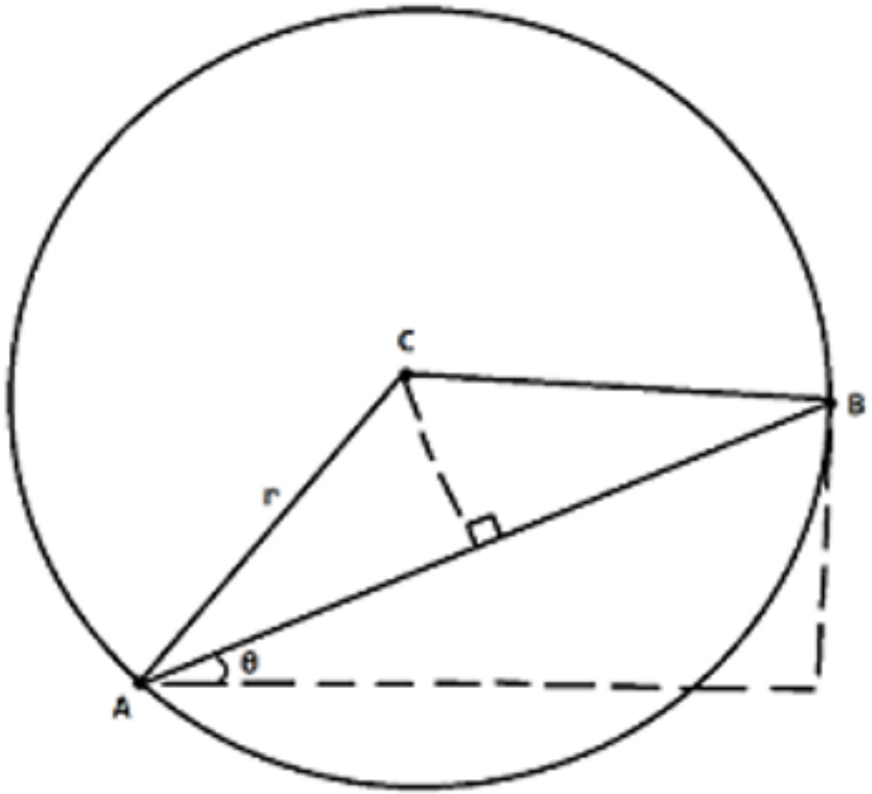
- How to calculate the distance between two points?
- Use simple math you learnt in your high school!
- Use trigonometry to get your angle between two lines.



$$c^2 = a^2 + b^2$$



Lab 1: Level 3 Hint



Lab 1: `List` API

- You will need to use methods from the `List` API to help you with Level 5
- Simply put, `List` provides us with an abstraction to represent a list of items (as the name implies)
- API: https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/List.html
- An important part of programming is learning to read the API

```
import java.util.List; // List is not part of java.lang
```

```
jshell> class Point {
   ...> private int x;
   ...> private int y;
   ...> Point(int x, int y) {
   ...> this.x = x;
   ...> this.y = y;
   ...> }
   ...> }
|  created class Point

jshell> List<Point> points = List.of(new Point(1, 1), new Point(2, 2))
points ⟹ [Point@51565ec2, Point@482f8f11]

jshell> points.size()
$3 ⟹ 2

jshell> points.get(1)
$4 ⟹ Point@482f8f11

jshell> points[1]
|  Error:
|  array required, but java.util.List<Point> found
|  points[1]
|  ^-------^

jshell> points.get(points.size())
|  Exception java.lang.IndexOutOfBoundsException: Index: 2 Size: 2
|        at ImmutableCollections$AbstractImmutableList.outOfBounds (ImmutableCollections.java:205)
|        at ImmutableCollections$List12.get (ImmutableCollections.java:393)
|        at (#5:1)
```

Future lesson flow

1. Let me know if you're not coming and why; attending make-up tutorial.
2. Attendance QR code. *Please submit the necessary FET results!*
3. Login to plab **between 1000 - 1015hrs**.
4. Clarify doubts, tell me about the weather.
5. Start lesson
6. Code + Clarify doubts
7. Take mask-off photo