Pair Programming

What is Pair Programming?

- Two developers work together on one project
- One developer codes while the other reviews
- Co-responsibility for outputs
- Developers can switch roles when needed
- Effective for identifying bugs and design problems, and maintaining coding standards

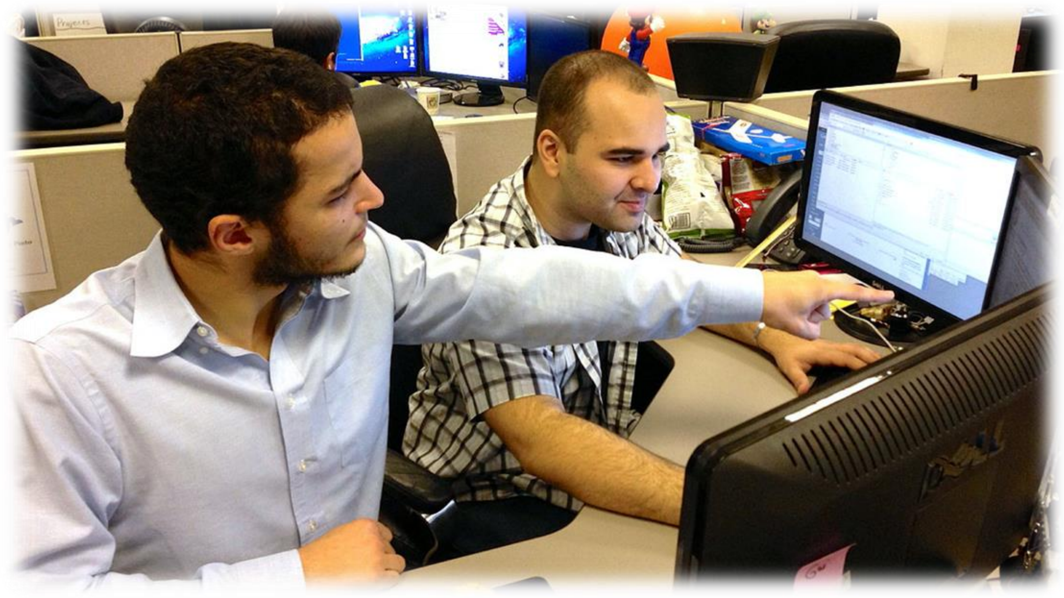
Image: By Kabren - Own work, CC BY-SA 3.0, [https://commons.wikimedia.org/w/index.php?curid=21903816]

BENEFITS OF PAIR PROGRAMMING

- **Increased productivity**
- **Increased confidence**
- **Cross-learning**
  - Reduced training costs and time
- **Multiple points of view**
  - Faster issue resolution
  - Fewer obstacles
- **Continuous code reviews**
  - Give feedback and reduce bugs more quickly
- **Shared responsibility**
  - Trust
  - Backups if a developer is on annual leave or falls ill

TYPES OF PAIRING

- Driver-Navigator
- Backseat navigator
- Tour guide
- Ping-pong
- Cross-functional
- Distributed

Pair programming types

- Expert-Expert
- Expert-Novice
- Novice-Novice
- Developer-Operations engineer

Driver-Navigator (most common style)

It is like when you negotiate a car trip through unfamiliar territory.

Driver

handles typing, save and load files, and other implementation issues.

Navigator

looks at other issues like

- checking for mistakes.
- does the code fit with the designed architecture?
  - are we duplicating code which exists elsewhere?
  - are we in a blind alley (code going nowhere)?

Backseat Navigator (Novice & expert)

Driver

Takes care of typing code and saving files

Navigator

Dictates tactical instructions.

- When to create a class/method or a new file.
- What to name a file or unit test or even a variable.

Tour Guide (Expert and new Novice)

It is like when you take a trip conducted by local tour guide.

The driver (expert), starts driving, and tells you about everything he/she is doing and tell the passengers (novice) about what they are seeing.

It may look like the passengers have a passive role, but they must take note and learn.

Roles can be flipped!
Let the novice do the work and even fail at the problem while the expert observes. After a while, the expert provides feedback and correction.

Pair programming types…

- **Ping-Pong: (good for TDD)**
  The first programmer writes a failing test and the other writes the code to pass it.

- **Cross-functional (Developer works with a hardware engineer)**
  Allows more time to work alone
  Ideal for developing new systems

- **Distributed (members of the team are in different geographical locations)**
  - Developers work together via a collaborative real-time editor, shared desktop or remote pair programming plugin
  - More tiring than traditional pair programming (more hours)
  - Tools used include *Mikogo*, Trellis and *Yuuguu*

LAB

PAIR PROGRAMMING PRACTICE

Use one of the Pair Programming methods we discussed.
You only need to come up with one solution between you –
but it must be a joint effort.
Think about how you could share code as you work together

There is a C# and Java starter projects but you will write all the code!
If you are using Python or JavaScript, you will need to start fresh.

Afterwards, we'll discuss...

- Which pair programming method(s) you tried
- What went well
- What didn't go so well