CS 455/555: Project Logistics

Project teams: Pair Programming

- *Pair programming* is an agile software development technique in which two developers work as a pair together on one workstation.

Project teams: Pair Programming

- *Pair programming* is an agile software development technique in which two developers work as a pair together on one workstation.
- The driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in. The navigator also considers the "strategic" direction of the work and considers improvements and likely future problems. The two developers switch roles frequently.
  https://en.wikipedia.org/wiki/Pair_programming

Project teams: Pair Programming

- *Pair programming* is an agile software development technique in which two developers work as a pair together on one workstation.
- The driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in. The navigator also considers the "strategic" direction of the work and considers improvements and likely future problems. The two developers switch roles frequently.
  https://en.wikipedia.org/wiki/Pair_programming
- *Lean Software Development: Building and Shipping Two Versions* by Kate Matsudaira. Communications of the ACM, Vol. 58 No. 12, Pages 56-58. (*Note*: This article will be accessible only on campus or if you are logged in to your Boise State account).
  http://cacm.acm.org/magazines/2015/12/194642-lean-software-development/fulltext

Find your buddy!

- Have you taken Operating Systems course?
- Have you taken Networks course (with network programming)?
- A solid understanding of multithreaded programming in any language.

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.
- C++: Google File System, YouTube, parts of Facebook, Twitter, Wikipedia

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.
- C++: Google File System, YouTube, parts of Facebook, Twitter, Wikipedia
- Go: Google, Docker, Kubernetes, Dropbox, Heroku, Netflix.

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.
- C++: Google File System, YouTube, parts of Facebook, Twitter, Wikipedia
- Go: Google, Docker, Kubernetes, Dropbox, Heroku, Netflix.
- Languages often used with a specific framework:
  - Node.js, Python (with Django), Ruby on Rails, PHP

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.
- C++: Google File System, YouTube, parts of Facebook, Twitter, Wikipedia
- Go: Google, Docker, Kubernetes, Dropbox, Heroku, Netflix.
- Languages often used with a specific framework:
  - Node.js, Python (with Django), Ruby on Rails, PHP
- *Functional Programming Languages*:

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.
- C++: Google File System, YouTube, parts of Facebook, Twitter, Wikipedia
- Go: Google, Docker, Kubernetes, Dropbox, Heroku, Netflix.
- Languages often used with a specific framework:
  - Node.js, Python (with Django), Ruby on Rails, PHP
- *Functional Programming Languages*:
  - Scala (JVM based): LinkedIn, Twitter, New York Times, Spark

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.
- C++: Google File System, YouTube, parts of Facebook, Twitter, Wikipedia
- Go: Google, Docker, Kubernetes, Dropbox, Heroku, Netflix.
- Languages often used with a specific framework:
  - Node.js, Python (with Django), Ruby on Rails, PHP
- *Functional Programming Languages*:
  - Scala (JVM based): LinkedIn, Twitter, New York Times, Spark
  - Erlang: Facebook Chat, WhatsApp

What Language to Use for the Backend?

- Java: Amazon, Netflix, eBay, LinkedIn, Google AdWords, GMail, Twitter, Hadoop, Hive, Zookeeper, Cassandra.
  - Major chunks of NY Stock exchange, LSX, CME etc. Trading systems, risk systems etc at basically every large bank.
- C++: Google File System, YouTube, parts of Facebook, Twitter, Wikipedia
- Go: Google, Docker, Kubernetes, Dropbox, Heroku, Netflix.
- Languages often used with a specific framework:
  - Node.js, Python (with Django), Ruby on Rails, PHP
- *Functional Programming Languages*:
  - Scala (JVM based): LinkedIn, Twitter, New York Times, Spark
  - Erlang: Facebook Chat, WhatsApp
  - Elixir: Pinterest

Java

- Used widely to build distributed systems so there are many supporting libraries available.

Java

- Used widely to build distributed systems so there are many supporting libraries available.
- Networking features are in the core of the language: support for TCP/IP, UDP/IP (including multicast), high-performance non-blocking I/O, HTTP / URx support, REST support, XML support, JSON support, and security features like SSL.

Java

- Used widely to build distributed systems so there are many supporting libraries available.
- Networking features are in the core of the language: support for TCP/IP, UDP/IP (including multicast), high-performance non-blocking I/O, HTTP / URx support, REST support, XML support, JSON support, and security features like SSL.
- Rich set of data structures are available in core libraries.

Java

- Used widely to build distributed systems so there are many supporting libraries available.
- Networking features are in the core of the language: support for TCP/IP, UDP/IP (including multicast), high-performance non-blocking I/O, HTTP / URx support, REST support, XML support, JSON support, and security features like SSL.
- Rich set of data structures are available in core libraries.
- Strong multithreaded performance. Java supports natively thread creation, locking primitives, wait/notify, multithreaded data structures, and other more advanced support for complex multi-threaded programming.

Java

- Used widely to build distributed systems so there are many supporting libraries available.
- Networking features are in the core of the language: support for TCP/IP, UDP/IP (including multicast), high-performance non-blocking I/O, HTTP / URx support, REST support, XML support, JSON support, and security features like SSL.
- Rich set of data structures are available in core libraries.
- Strong multithreaded performance. Java supports natively thread creation, locking primitives, wait/notify, multithreaded data structures, and other more advanced support for complex multi-threaded programming.
- Supports serialization of objects natively. This is needed to send complex data structures over the network.

Java

- Used widely to build distributed systems so there are many supporting libraries available.
- Networking features are in the core of the language: support for TCP/IP, UDP/IP (including multicast), high-performance non-blocking I/O, HTTP / URx support, REST support, XML support, JSON support, and security features like SSL.
- Rich set of data structures are available in core libraries.
- Strong multithreaded performance. Java supports natively thread creation, locking primitives, wait/notify, multithreaded data structures, and other more advanced support for complex multi-threaded programming.
- Supports serialization of objects natively. This is needed to send complex data structures over the network.
- Java APIs are used widely in industry. The platform is mature and evolving with time.

References

- Programming languages used in the most popular websites. https://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites
- *Lean Software Development: Building and Shipping Two Versions* by Kate Matsudaira. Communications of the ACM, Vol. 58 No. 12, Pages 56-58.