

Open Source Software Project Development

Dr. T.Y. Wong

Course Information

Course Aim

- **Not to teach you how to open your source.**
- Then, what do we teach?
 - The techniques in writing “real” software.
 - That’s to say “*most of your previous assignments are just **TOYS***”.
 - Double-quoting “real” means that, sorry, we will be implementing **ADVANCED TOYS**.
 - Trendy open-source projects.
 - We will try to break those trendy projects into pieces.
 - Then, study them piece-by-piece.

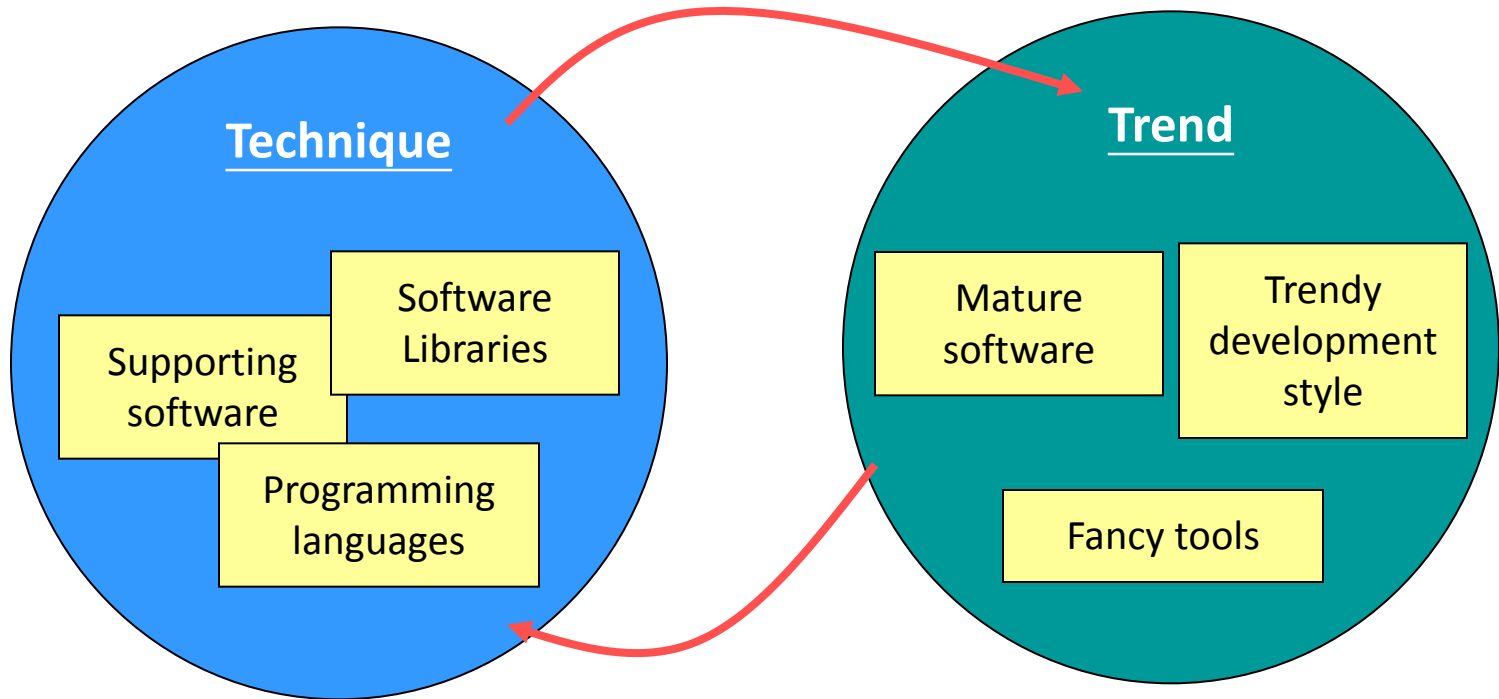
Course Materials

- No books. Only lecture notes.
 - also using Wiki, YouTube, etc.
- No official programming languages.
 - But, we'll use many languages.

Assumed you know	C, Basic SQL.
Taught	Lectures: JavaScript, PHP, Perl. Tutorials: HTML (tutorials only).
Used in lectures	C, JavaScript, Perl, Python, HTML, PHP, SQL.
Used in assignments	PHP, Perl, HTML, SQL, JavaScript.
Used in projects	Whatever you like!

Open-source development?

- Open-source development is a feedback loop...

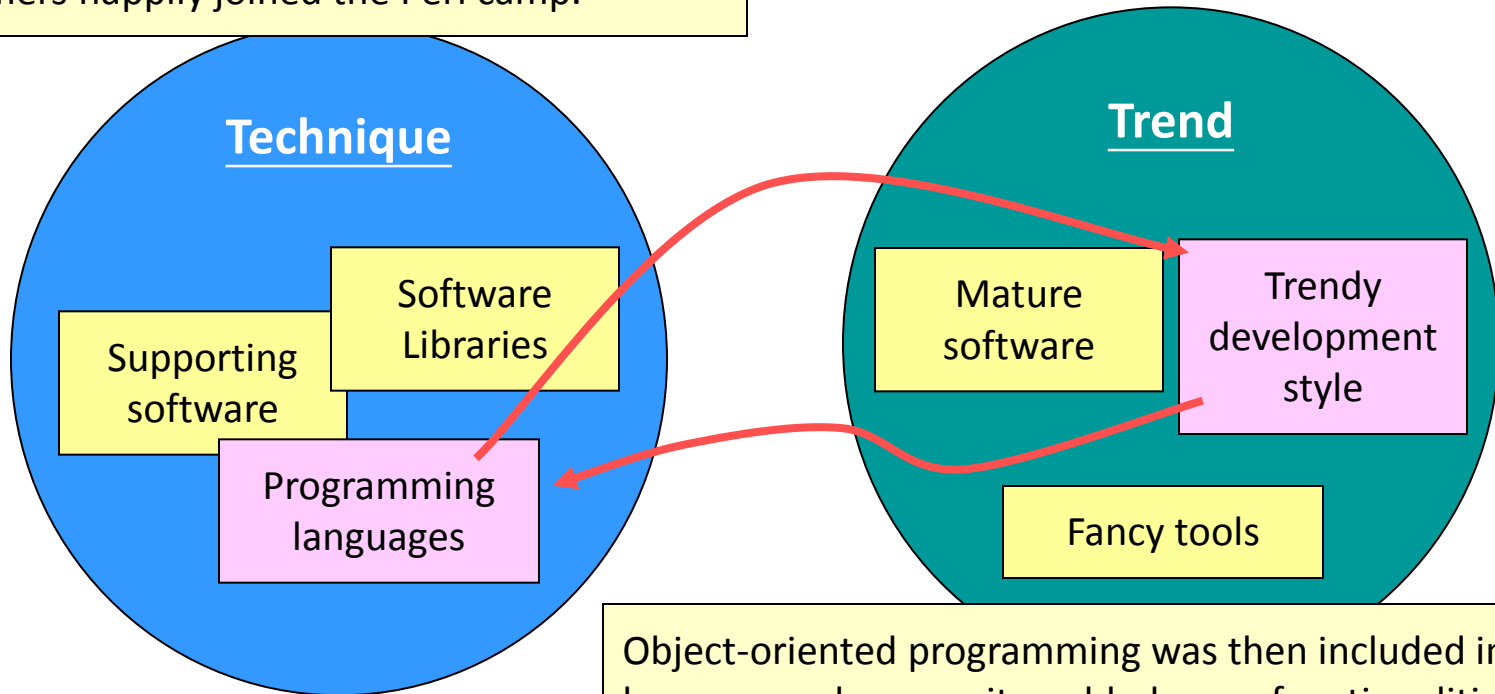


Open-source development?

- An example...

E.g., during late 1990s, using Perl is nearly a **MUST** for CGI developments. It is a trend that C and C programmers happily joined the Perl camp.

So, how many C programmers left?



Object-oriented programming was then included in to the language as because it enabled more functionalities to the CGI applications.

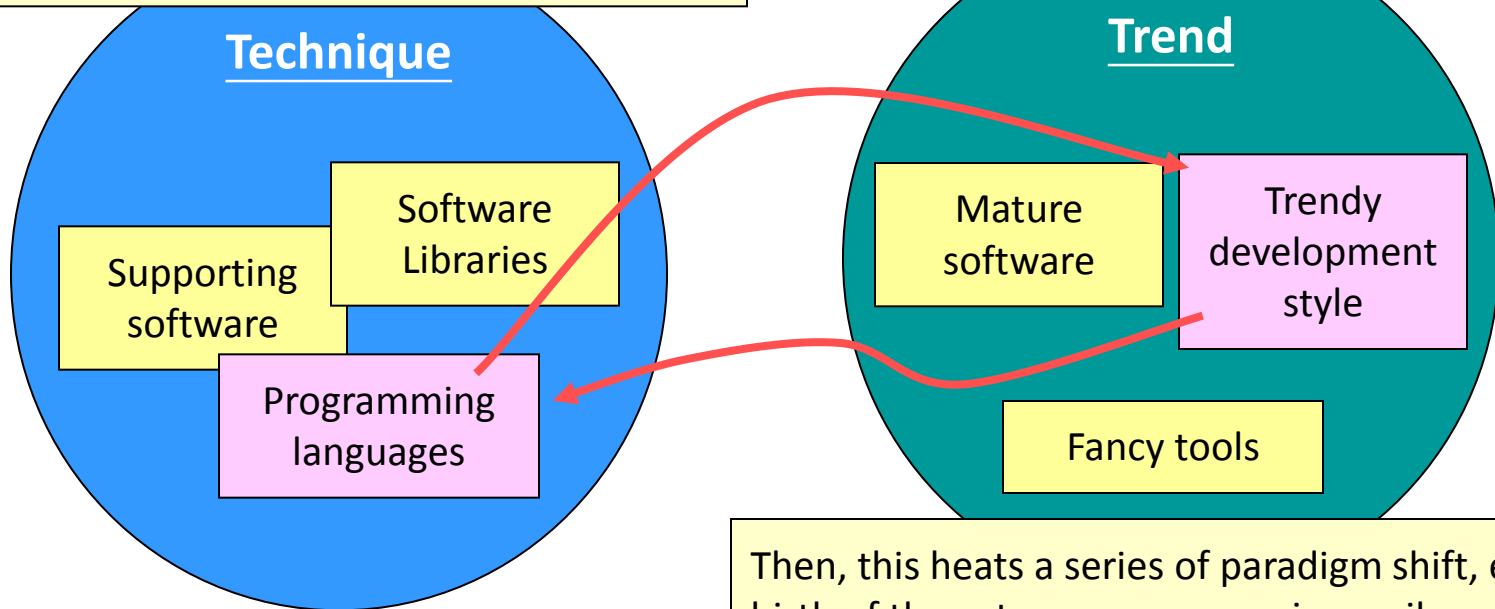
Open-source development?

- An example...

Then, people think that Perl is too bulky and **brain-damaging**...

This pushes the rapid adoption on the language Python since **Python is more versatile than Perl**. Many Perl programmers then join the Python camp.

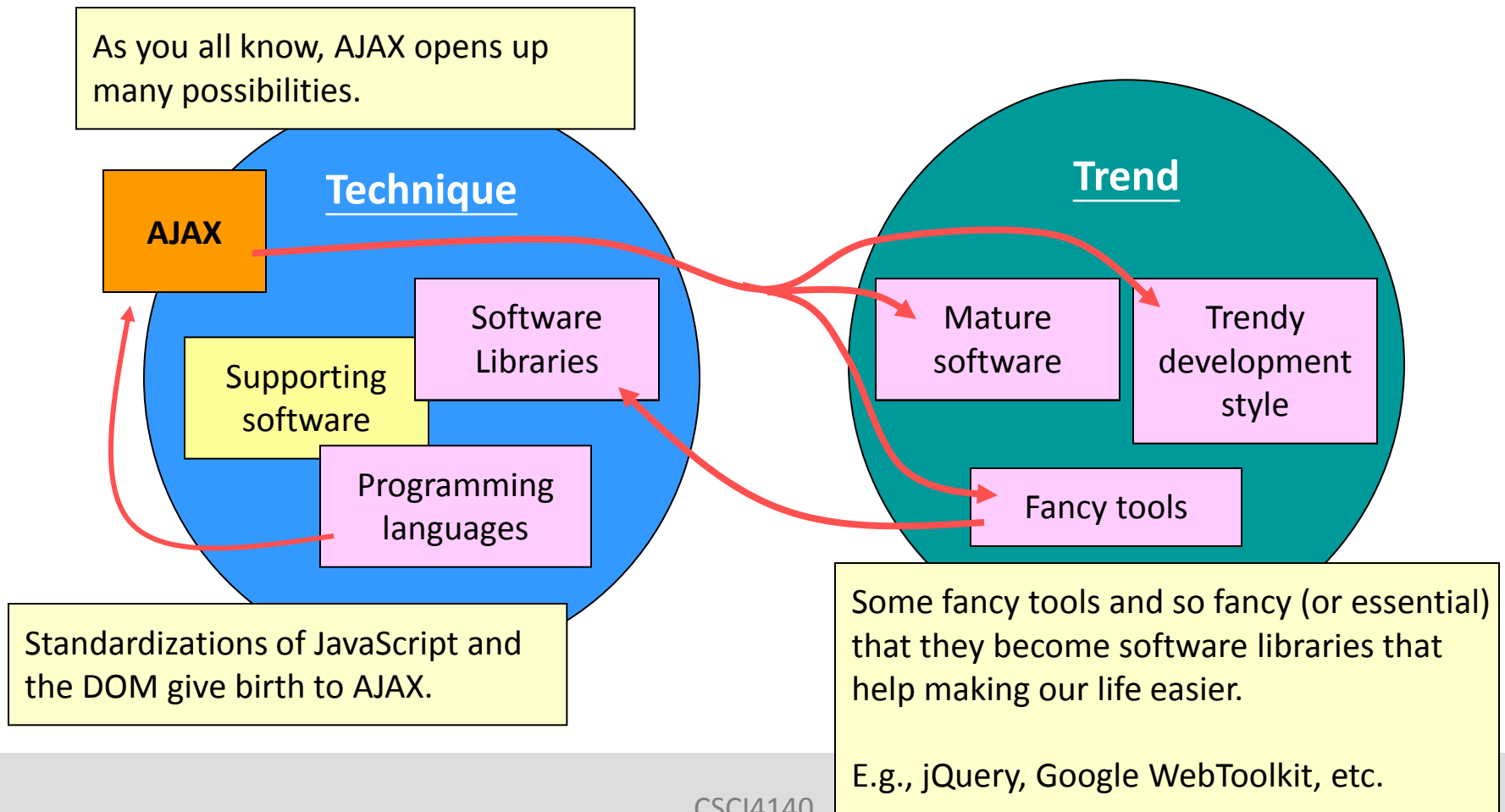
Again, how many C programmers left?



Then, this heats a series of paradigm shift, e.g., the birth of the extreme programming, agile development, etc...

Open-source development?

- The open source development community is very active and always contributes new ideas...



Course Objectives

- Most of the open-source software projects are **to make your life easier**.
 - No matter you're a programmer or not.
 - So, we're going to understand what are the techniques that make our life easier.
- Note that many open-source software is adopted into the enterprise mainstream software.
 - So, it is important to know the ways that make people **feeling safe** to use the software.

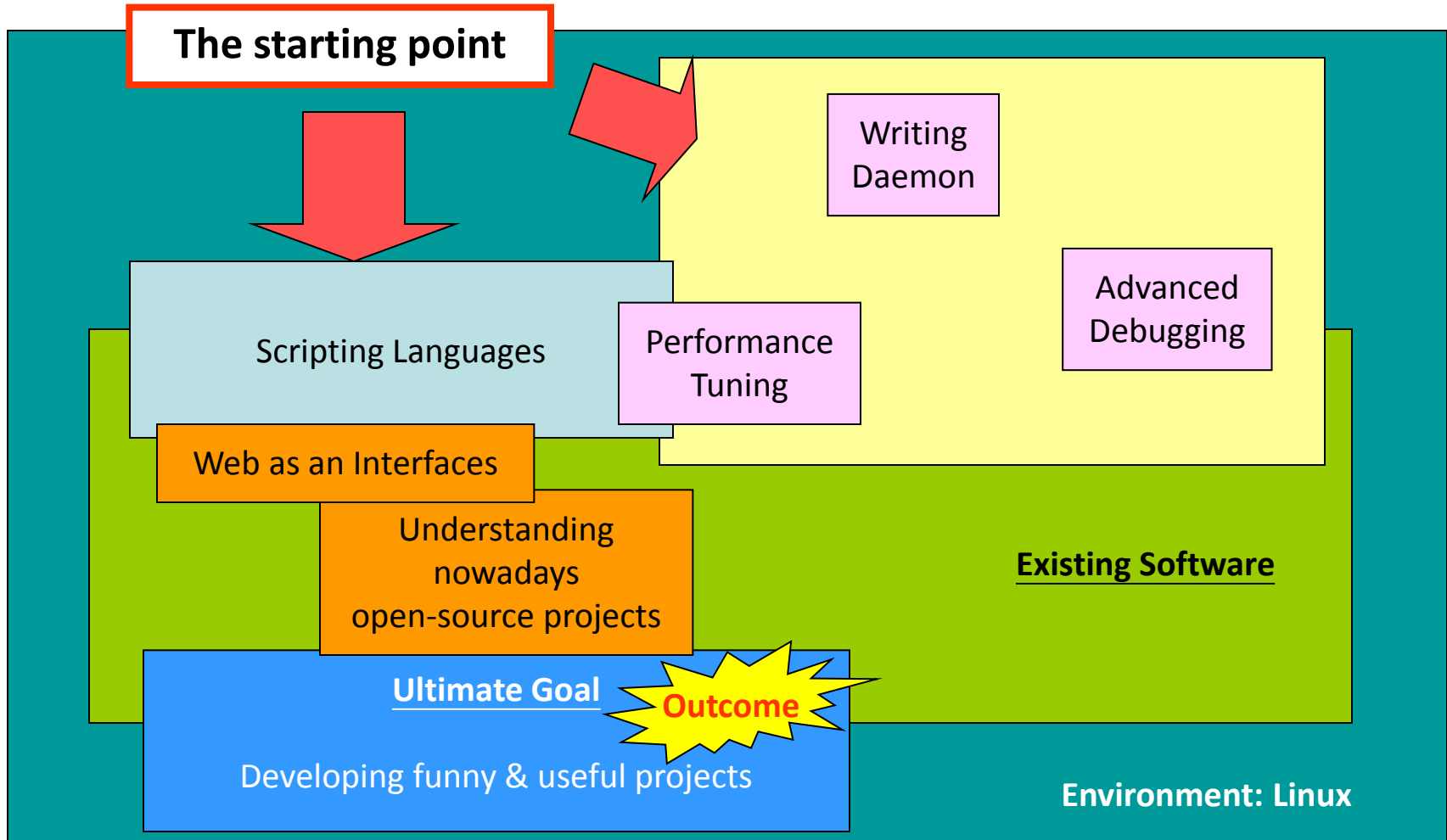
Course objectives

Software engineering courses	Our course
Focus on the formal steps in developing software, independent of the technologies.	Focus on how to leverage nowadays technologies to develop software.
Focus on the requirements and the outcomes.	The requirements may be coming from the lead programmer himself and are usually subject to change.
Focus on the testing of the software based on the requirement specification(s).	Most of the time, there is only a goal with a loosely written specification. So, we are going to learn how to write robust software.

A formal course

A casual course

Approach?



Assessment

- Two assignments – 30%.
- One free-style project – 35%.
 - If the project can be helpful to your FYP, dreams, etc, then it'll be great!
 - Restrictions:
 - You have to discuss with the lecturer and the tutor.
 - Then, the lecturer and the tutor will **estimate the score that the proposed project is worth**.
 - Avoid doing any direct copying.
- One final examination – 35%.

Let's Rock!