

CSCI 334:
Principles of Programming Languages

Lecture 19: Packages & Testing

Instructor: Dan Barowy
Williams

Topics

Package frameworks

Unit testing

Your to-dos

1. Lab #10, **due Sunday 12/3**
2. If you found lab #9 difficult, come see me!
Office hours today 4-5pm.

Final project timeline

- ~~1. Project proposal (Lab 8), **due Sun 11/12**~~
- ~~2. Minimally working version (Lab 9), **due Sun 11/19**~~
3. Language specification doc (Lab 10), **due Sun 12/3**
4. Mostly working version (Lab 11), **due Sun 12/10**
5. Project + video presentation (Lab 12), **due Sun 12/17**

Language Package Framework

A **language package framework** is a repository of **software libraries**, together with a **distribution mechanism**, to simplify finding and incorporating **third-party software** into your code.

A good package framework makes using a language more **productive** and **fun!**

History

The first widely-known package framework was **CTAN**, the **Comprehensive TeX Archive Network**, started in 1991 for the TeX language. Files were originally distributed using the FTP protocol.

Many languages followed...

Popular Package Frameworks

Java: Maven

.NET: NuGet

Python: pip

Ruby: gem

Rust: crates

Go: cargo

Tons more!

<https://www.nuget.org/>

(code)

Unit testing

Unit testing is a quality-assurance method designed to find bugs before software ships. A **unit test** consists of **test code** written to exercise the functionality of a **unit** of code **in isolation**. For example, in functional code, a unit is often thought of as a **module**, **function**, or **primitive** operation.

Note that unit testing is usually **not sufficient** to determine the correctness of code!

Popular Unit Test Frameworks

Java: JUnit

.NET: MsTest or NUnit

Python: built in!

Ruby: built in!

Rust: built in!

Go: built in!

Tons more!

https://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

Regression

Unit testing helps prevent **regressions**.

A **regression** is a kind of software bug where a feature that **worked earlier stops working**.

Because of their expressive and compositional nature, regressions are very common in language development. By developing a unit test suite alongside your implementation, you will **save time** and write **better code** because you will be able to detect regressions early.

(demo)

Test-driven Development

Test-driven development is a software development process that emphasizes **writing a test** for a planned feature **before implementing** a planned feature.

Procedure:

1. **Add a test.**
2. **Run all tests.** The new test should **fail**.
3. **Write the simplest implementation** that should pass the new test.
4. **Run all tests.** The new test should **pass**. If it does not, go to 3.
5. **Rewrite as needed** to enhance readability or maintainability.

Recap & Next Class

This lecture:

Packages

Unit testing

Next lecture:

Scope