



Course Syllabus

Course Code	Course Title	ECTS Credits
COMP-142	Software Development Tools for Data Science	6
Prerequisites	Department	Semester
None	Computer Science	Spring
Type of Course	Field	Language of Instruction
Required	Data Science	English
Level of Course	Lecturer(s)	Year of Study
1 st Cycle	Dr Harald Gjermundrød	1 st
Mode of Delivery	Work Placement	Corequisites
Face-to-face	N/A	COMP-113

Course Objectives:

The main objectives of the course are to:

- gain practice and experience in the software development process and tools used for data science
- introduce and provide experience in using tools for source code version management and source code documentation practices
- familiarize students with tools for software bug tracking in order to gain experience in using such tools
- cover in detail the full life-cycle of software development: commit code, test and submit bug reports, provide patches and patch the source code
- introduce and use tools to develop and run unit tests
- introduce and use tools for builds and installers for various platforms and/or architecture
- introduce and use tools for source code and application profiling.

Learning Outcomes:

After completion of the course students are expected to be able to:

1. apply the full cycle of software development using appropriate tools at the various stages
2. use tools effective for source code version management and team development
3. use tools effective for source code documentation
4. use bug-tracking tools for application development

5. develop and run unit tests as part of the development cycle
6. create builds and installers for a software product
7. effectively use profiling tools both for code and application profiling.

Course Content:

1. The Software Development Process
 - a. From Problem Analysis to Code and then to Software
2. Integrated Development Environments
 - a. Compiling
 - b. Linking
 - c. Debugging
 - d. Variable Monitoring
3. Software Libraries
 - a. Libraries for Statistics, File IO and String Processing (e.g., StatLib, Boost, Eigen)
 - b. Library/Dependency Management
4. Source Code Version Tools
 - a. How to create, check-in, check-out, source code from a version management tool and compare different source code versions.
 - b. Collaborative Programming
5. Documentation
 - a. Commenting
 - b. Automated Documentation Tools (e.g., Doxygen)
 - c. Documentation Practices
6. Testing and Error Handling
 - a. Unit Tests
 - b. Test Suites
 - c. How to develop unit tests and integrate them into the software development lifecycle.
7. Source code bug tracking tools
 - a. How to report bugs, report feature requests, accept bug reports, browse current reports and amend them.
 - b. How to create patches (using the development environment), attach them to current bug reports (using the bug tracking tool), apply and verify patches, and check in the patched code (using the version management tool).
8. Build and Release Tools
 - a. Introduction to build and release systems.
 - b. How to wrap up the builds and encapsulate them into installers.
9. Profiling Tools
 - a. Code Profiling (e.g., execution time)

b. Application Profiling (e.g., resource utilization, memory leaks)

Learning Activities and Teaching Methods:

Lectures, Lab Tutorials, Practical Exercises, and Assignments

Assessment Methods:

Final Exam, Assignments, and Quizzes

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Introduction to Programming with C++, 3 rd Ed.	Daniel Liang	Pearson	2013	0-13-325281-7
Software Development Lifecycle: Lecture Notes	Dr. Harald Gjermundrød	Available on course page		

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Practical Development Environment	Matthew B. Doar	O'Reilly	2005	0-596-00796-5
Version Control with Subversion, 2 nd Ed.	Pilato, Collins-Sussman, Fitzpatrick	O'Reilly	2008	0-596-51033-6