Title	Scientific Software Development
Type of educational activity/teaching format	Mixture of Lecture/Seminar/Lab Exercise, online
Offering lecturer and/or responsible contact person Name, surname, position, affiliation, contact information	Dr. Inga S. Ulusoy, Research Software Engineer, Scientific Software Center, Interdisciplinary Center for Scientific Computing, Heidelberg University, inga.ulusoy@uni-heidelberg.de
Other lecturers (if involved) * Name, surname, position, affiliation	
Start date - end date and duration	1.3.22-10.3.22; every Tuesday, Wednesday, Thursday from 10-1 (6 sessions)
Faculty	(Mathematics and Computer Science/Engineering, but in fact for students of all faculties)
Subject *	Programming and Software Development
Available places (for students from 4EU+ universities)	20
Short description of the course (4-5 sentences)	This course encompasses how to develop scientific software in a way that enables reproducible research and sustainable software. Sustainable scientific software leaves room for further, subsequent implementations and method development, and saves time and effort in the long run; in addition to adhering to good scientific practice through making results reproducible. This course is aimed at researchers, doctoral and master students who develop scientific software to carry out their research. The course consist of both an asynchronous and a synchronous part. See https://ssciwr.github.io/sustainable_development_course/
Workload / Credits	3 FÜK („Fachübergreifende Kompetenzen" – extracurricular credits)
Target group-level (BA, MA and/or PhD students)	PhD students, MA students of all disciplines
Languages of instruction	English, Python
Learning content	The course covers version control using git, data exploration using jupyter notebooks, planning of scientific software, working as a team, documentation using sphinx, testing with pytest, continuous integration, and publishing a python package.
Intended Learning Outcomes	At the end of the course, the students will be able to develop their own scientific software sustainably and generate reproducible results, in accordance with best scientific practices.
Assessment methods and criteria	Active participation in the live sessions, implementation of the subject matter into a GitHub repository.
4EU+ Flagship	Flagship 3
4EU+ Transversal skills/shared competencies –multilingualism –data literacy –critical thinking –entrepreneurship	Data literacy, critical thinking, entrepreneurship

–social engagement	
Requirements for participation * (are proofs required)	Very basic knowledge of programming, ideally in Python
Composition of the final grade *	
Application process *	Subscribe at https://ssc.iwr.uni-heidelberg.de/form/scientific-software-development-
Application deadline *	February 15th
Link to local course catalogue *	
URL and Key for moodle course *	Not required