

Assignment Brief Form

Course information from DRPS

Course acronym	ACP
Course code	INFR11245 / INFR11249
Credits	10
Course Organiser(s)	Michael Glienecke
Learning Outcomes <i>Paste in from DRPS</i>	<p>On completion of this course, the student will be able to:</p> <ol style="list-style-type: none"> 1. implement the basic building blocks of cloud programming like containerization, micro-services, event-processing and cloud data storage using standard service offerings by the main cloud providers in combination with several communication protocols 2. compare / contrast the differences between architecture styles like IaaS, PaaS, CaaS, SaaS, etc 3. summarize cloud service offerings by the main providers and evaluate the differences 4. contrast different contemporary cloud runtime environments 5. differentiate the need and structure of Continuous Integration / Continuous deployment (CI/CD)-chains in cloud programming

Mapping of Learning Outcomes onto Assessment

Please indicate which Learning Outcomes are assessed by which assessment components by entering a "+" in the cells of the table below.

	LO1	LO2	LO3	LO4	LO5
CW1	++		+		
CW2	+++		+++		+
CW3 (student choice)	++				
CW4 (video presentation)		++	+	++	+

Brief for each coursework

CW1

Assignment name	Programming Assignment
<i>This name should correspond to the Coursework Name in PIP</i>	
Task overview	The student must implement a rudimentary Java REST-service with some 4-5 pre-defined endpoints which are re-used in CW2. Simple logic, mostly to check for setup, REST-basics and simple testing
Assessment criteria	Each task carries points. For minor deviations (e.g. slightly incorrect results) a penalty is taken; otherwise, 0 points for the given task
Submission instructions	Docker-image plus all source-codes as Learn submission
Guidance on size of submission	N/A
Penalties for overlong submissions	N/A
Feedback procedure	E-Mail with individual auto-marker results (complete transcript), discussion in class about the most common problems and Q&A / individual sessions if needed
Support arrangements	Tutorials, Q&A, TA, piazza
Marking and moderation procedure	Docker-image is auto-marked producing a result record with points (uploaded to Learn) The auto-marker produces detailed information per task and highlights problems For moderation students reach out to the CO and then discuss the issues.



CW2

Assignment name	Programming Assignment
<i>This name should correspond to the Coursework Name in PIP</i>	
Task overview	Like CW1, just more elaborate (5-6 endpoints and more complex implementation logic).
Assessment criteria	Each task carries points. For minor deviations (e.g. slightly incorrect results) a penalty is taken; otherwise, 0 points for the given task
Submission instructions	Docker-image plus all source-codes as Learn submission
Guidance on size of submission	N/A
Penalties for overlong submissions	N/A
Feedback procedure	E-Mail with individual auto-marker results (complete transcript), discussion in class about the most common problems and Q&A / individual sessions if needed
Support arrangements	Tutorials, Q&A, TA, piazza
Marking and moderation procedure	Docker-image is auto-marked producing a result record with points (uploaded to Learn) The auto-marker produces detailed information per task and highlights problems For moderation students reach out to the CO and then discuss the issues.



CW3

Assignment name	Student choice implementation <i>This name should correspond to the Coursework Name in PIP</i>
Task overview	The student can implement anything in relation to the contents of ACP in either Java, Python, JavaScript, react, Go or Rust. This could be a UI, some complex pre- / post-processing system, some data pipeline, etc. MUST (!) utilize a free service from any cloud provider Student's creativity is the limit
Assessment criteria	The main criterion for marking is the PDF, the sources only serve as evidence or investigation point. <ul style="list-style-type: none">• Innovation / Idea / Benefit (4)• Execution / Implementation (8)• Completeness (4)• Style (4)
Submission instructions	The student must submit all sources + necessary docker images plus a description PDF which explains the implementation: Why, for whom, what problem shall be solved / mitigated, details of implementation. AI is allowed for planning
Guidance on size of submission	ZIP-file + PDF in Learn PDF not more than 1000 words / 4 pages with diagrams as guidance.
Penalties for overlong submissions	More than 50% excess will incur a 25% deduction of marks
Feedback procedure	Feedback in Learn and based on the rubrics
Support arrangements	Tutorials, piazza, Q&A, TA, direct contact
Marking and moderation procedure	Marked by 3 markers in parallel to avoid bias (medium is taken)



CW4

Assignment name <i>This name should correspond to the Coursework Name in PIP</i>	Video presentation of the student implementation plus additional topics
Task overview	The student provides a 8-11 min video where the implementation is presented alongside background, motivation, target audience. In addition, several predefined questions regarding learning outcome 3 / 4 / 5 must be presented and set in context to the chosen implementation
Assessment criteria	<ul style="list-style-type: none">- Background elaboration / target audience (10%)- Problem statement (10%)- Explanation of the student's choice (20%)- Presentation (20%)- Pre-defined questions (20%)- Elaboration of relation to student's implementation (20%)
Submission instructions	Upload to Learn as a video. If the video is too long -> 10% deduction for every minute over 11;
Guidance on size of submission	N/A (8 - 11 min)
Penalties for overlong submissions	10% deduction for every minute over 11;
Feedback procedure	In Learn based on the Rubrics
Support arrangements	Tutorials, piazza, Q&A, TA, direct contact
Marking and moderation procedure	Marked by 2 persons (usually CO + TA) to avoid bias (arithmetic mean is taken)