

## Course: Strategien kompetitiver Programmierung (IDL)

| General information         |  |
|-----------------------------|--|
| <b>Course Name</b>          | Strategien kompetitiver Programmierung (IDL)<br>Strategies for competitive programming (IDL) |
| <b>Course code</b>          | COCO-IDL   |
| <b>Lecturer(s)</b>          | Oenings, Hendrik (hendrik.oenings@haw-kiel.de)   |
| <b>Occurrence frequency</b> | Regular  |
| <b>Module occurrence</b>    | In der Regel im Sommersemester   |
| <b>Language</b>             | Deutsch  |

| Qualification outcome   |  |
|---|--|
| <i>Areas of Competence: Knowledge and Understanding; Use, application and generation of knowledge; Communication and cooperation; Scientific self-understanding / professionalism.</i>  |  |
| Die Studierenden<br>- kennen grundlegende Algorithmen und Datenstrukturen.<br>- verstehen die Bedeutung von Laufzeit-/Speicherkomplexität.  |  |
| Die Studierenden<br>- können gegebene Problemstellungen auf bekannte algorithmische Methoden reduzieren und diese anpassen.<br>- können den Umfang eines Problems einschätzen und beurteilen, welche algorithmischen Strategien effizient einsetzbar sind.<br>- können ungefähr einschätzen, wie aufwändig die Lösung eines Problems in Bezug auf Laufzeit und Implementierungsaufwand ist.   |  |
| Die Studierenden<br>- können im Team mit anderen ihre Ideen kommunizieren und erarbeiten.<br>- können aus natürlichsprachlichen Texten und Beispielen abstrahieren und Problemstellungen erkennen.  |  |
| Die Studierenden<br>- können selbstständig komplexe Aufgabenstellungen einschätzen und Lösungsstrategien entwickeln.<br>- können ihre Ergebnisse reflektieren und bewerten und Grenzen ihrer eingesetzten Methode in Bezug auf Speicherbedarf und Laufzeit in der Praxis einschätzen.<br>- können bei komplexen Aufgabenstellungen beurteilen, welche Lösungsstrategie(n) unter Berücksichtigung sowohl der Problemgröße und Laufzeit als auch des Implementierungsaufwands einzusetzen sind. |  |

| Content information |  |
|---------------------|--|
| <b>Content</b>      | - Bestimmung der Laufzeitkomplexität eines Algorithmus<br>- Datenstrukturen (dynamisches Array, Set, Map, ...)<br>- Backtracking<br>- Greedy-Algorithmen<br>- Dynamische Programmierung<br>- Range-Queries<br>- Graph-Algorithmen (DFS/BFS, Shortest path, Spanning tree, Ford-Fulkerson, ...)<br>- String-Algorithmen |
| <b>Literature</b>   | A. Laaksonen: Competitive Programmer's Handbook<br>( <a href="https://cses.fi/book/book.pdf">https://cses.fi/book/book.pdf</a> )   |

| Teaching format of this course |            |
|--------------------------------|------------|
| <b>Teaching format</b>         | <b>SWS</b> |
| Lehrvortrag + Übung            | 2          |

| <b>Examinations</b>               |     |
|-----------------------------------|-----|
| <b>Ungraded Course Assessment</b> | Yes |

| <b>Miscellaneous</b> |   |
|----------------------|---|
| <b>Miscellaneous</b> | <p>Für die Teilnahme sind Programmierkenntnisse hilfreich, aber nicht erforderlich.</p> <p>Studierende, welche diese Lehrveranstaltung als IDL-Veranstaltung einbringen, können nicht gleichzeitig das Modul COCO einbringen.</p> <p>Bei erfolgreicher Teilnahme an dieser Lehrveranstaltung erfolgt eine Anrechnung bei den damit verknüpften Modulen (z. B. WIL1, WIL2) mit 2,5 Leistungspunkten.</p> |