

<b>Module title</b> Advanced object-oriented software design			
<b>Module code</b> Tbd.	<b>Level</b> Bachelor (B.Sc.)	<b>ECTS credits</b> 5	<b>Duration</b> 2 weeks block course + virtual lectures
<b>Module instructor</b> Bertram Haskins, Nelson Mandela University	<b>Lecture type</b> Lectures + Guided Tutorial Sessions	<b>Prerequisite(s)</b> Java programming skills	<b>Grading</b> Tbd.
<p><b>Objectives</b> The purpose of this module is to instill students with knowledge of a wide variety of software design patterns, Object Oriented approaches to the implementation of software design patterns (in Java and / or related toolsets) and the real world uses of software design patterns.</p> <p><b>Course Outcomes.</b> By the end of the course, students will be able to:  <b>Knowledge &amp; Understanding:</b> a) Demonstrate an understanding of the theory and principles of selected design patterns; b) Choose when design patterns are applicable in the software development cycle; c) Identify aspects of software systems which are resistant to change; d) Communicate in a shared object-oriented vocabulary.  <b>Skills &amp; Abilities:</b> a) Analyse a complex system; b) Identify areas of tight-coupling; c) Model an object-oriented system from a design pattern view, by means of ERDs; d) Write code without unnecessary dependencies;</p> <p>The course consists of the series of lectures, interspersed with guided tutorials. The tutorials will apply the techniques introduced in the lectures. Having learned and practised the techniques on small examples, students will participate in a project which requires the implementation of selected design patterns to address provided requirements in a decoupled fashion. This will constitute the single assignment for the module.</p>			
<p><b>Content</b></p> <ol style="list-style-type: none"> <li>1. <b>Introduction to Design Patterns</b> <ol style="list-style-type: none"> <li>1.1. Overview of design patterns</li> <li>1.2. Types of design patterns</li> <li>1.3. Pattern Catalogs</li> <li>1.4. Anti-patterns</li> </ol> </li> <li>2. <b>Behavioral Patterns</b> <ol style="list-style-type: none"> <li>2.1. Template Method</li> <li>2.2. Command</li> <li>2.3. Chain of Responsibility</li> </ol> </li> <li>3. <b>Structural Patterns</b> <ol style="list-style-type: none"> <li>3.1. Adapter</li> <li>3.2. Façade</li> <li>3.3. Proxy</li> </ol> </li> <li>4. <b>Creational Patterns</b> <ol style="list-style-type: none"> <li>4.1. Prototype</li> <li>4.2. Builder</li> </ol> </li> <li>5. <b>Compound Patterns</b> <ol style="list-style-type: none"> <li>5.1. MVC</li> </ol> </li> </ol>			
<p><b>Textbook/teaching material</b></p> <ul style="list-style-type: none"> <li>• Java programming language</li> <li>• Eric Freeman. Head First Design Patterns. O'Reilly</li> <li>• Course notes</li> </ul>			

Note: this is not the official course descriptor according to the "Studien- und Prüfungsordnung" (SPO)