Form 20  
Heriot-Watt University - Course Descriptor Template (RAY)

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Advanced Software Engineering</th>
<th>School</th>
<th>Mathematical &amp; Computer Sciences</th>
<th>On or Off-Campus</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-ordinator</td>
<td>M Lones</td>
<td>SCQF Level</td>
<td>Course Code</td>
<td>F21AS</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>Semester</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

1. **Pre-requisites**  
Knowledge of Java programming and software engineering at undergraduate level

2. **Linked Courses**  
(FSpecify if synoptic)  
- F21SF Software Engineering Foundations

3. **Excluded Courses**  
None

4. **Replacement Course**  
Code:  
Date Of Replacement:
5. **Availability as an Elective**  
Yes [ ] No [ √ ]

6. **Degrees for which this is a core course**  
Mandatory course for MSc/PGD Software Engineering, Computing  
Optional course for various MSc programmes in Computer Science

7. **Aims**  
- To consolidate proficiency in imperative programming and software development
- To further develop object-oriented programming and object-oriented design methods
- To provide knowledge of simple data structures and algorithms
- To introduce concurrent programming techniques
- To instil understanding of the concepts and benefits of advanced software engineering methods
- To give further practical experience of the use of UML in software engineering
- To give practical experience of developing a substantial software engineering team project
- To enable the deployment of patterns in software engineering

8. **Syllabus**  
Data structures: stacks, queues, lists, priority queues, binary trees  
Algorithms: searching (linear and binary) and sorting  
Advanced object-oriented design techniques  
Thread-based programming: thread creation and interaction, shared variables and synchronisation  
Methodologies in software engineering practice; Unified Modelling Language; design patterns;  
Project planning and management in software engineering;  
Comparison of agile and plan-driven approaches
### Course Title
Advanced Software Engineering

### School
Mathematical & Computer Sciences

<table>
<thead>
<tr>
<th>Course Title</th>
<th>School</th>
<th>SCQF Level</th>
<th>Course Code</th>
<th>Semester</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Software Engineering</td>
<td>Mathematical &amp; Computer Sciences</td>
<td>11</td>
<td>F21AS</td>
<td>2</td>
<td>15</td>
</tr>
</tbody>
</table>

### Course Co-ordinator
M Lones

### Learning Outcomes (HWU Core Skills: Employability and Professional Career Readiness)

#### Subject Mastery
- Understanding, Knowledge and Cognitive Skills
- Scholarship, Enquiry and Research (Research-Informed Learning)

- Skill in the use of UML notation and translation of UML designs to working programs
- Understanding of basic data structures and algorithms and ability to critically evaluate their appropriateness and limitations for a range of moderately complex problems.
- Demonstration of skill in design and implementation of practical GUI based and theaded applications
- To demonstrate a critical understanding of modern software engineering practice and be able to evaluate the strengths and weaknesses of current software engineering methods and techniques
- To be able to choose appropriate metrics to measure software quality and quantity in a modern software engineering environment
- To be able to choose a suitable software development environment and development methodology for specific software development tasks and justify the choice

#### Personal Abilities
- Industrial, Commercial & Professional Practice
- Autonomy, Accountability & Working with Others
- Communication, Numeracy & ICT

- Appreciation of use of methodology to ground system analysis, design and development
- Understanding of different programming paradigms and their inter-relation
- Practice in working in a group, choosing a methodology, reaching a consensus, and working with others to a deadline
- Taking responsibility for own work, taking responsibility in the development of resources, critical reflection on development process and work undertaken by self.
- Effective appreciation of professional standards in modern software engineering practice.
- Showing initiative, creativity and team working skills in collaborative software development

### Assessment Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Duration of Exam (if applicable)</th>
<th>Weighting (%)</th>
<th>Synoptic courses?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exam</td>
<td>2 hours</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Coursework</td>
<td></td>
<td>50%</td>
<td></td>
</tr>
</tbody>
</table>

### Re-assessment Methods

- Exam: 2 hours

### Date and Version

- Date of Proposal: 1 February 2011
- Date of Approval by School Committee: 9 February 2011
- Date of Implementation: September 2011
- Version Number: 2.0