

## School of Informatics Course Proposal Form (version: May 2021)

Please see Page 2 for instructions on which parts of this form to complete, whom to consult with to avoid unnecessary effort, and where to send the completed form.

Proposer(s): Stuart Anderson

Date: 08 Mar 2022

### Cover page: Basic permanent course information

Unless otherwise noted, items in this section are entered into EUCLID and **cannot** be changed without creating an entirely new course.

<b>Course Name</b>	Software Testing
<b>Is this an EPCC course?</b>	<input checked="" type="checkbox"/> No (default) <i>(If you don't know what EPCC is, this is the right choice.)</i> <input type="checkbox"/> Yes <i>(If so, leave Course Acronym blank, to be filled in by ITO as EPCC/&lt;number&gt; for Theon and our Sortable List.)</i>
<b>Course Acronym</b> <i>(used only School-internally)</i>	ST
<b>SCQF Credit Level and Normal Year Taken</b>	<p><b>Standard options for Informatics courses:</b></p> <p><input type="checkbox"/> Level 8/Year 1</p> <p><input type="checkbox"/> Level 8/Year 2</p> <p><input checked="" type="checkbox"/> Level 10/Year 3 (also available in Year 4). <i>[In practice, most level 10 courses have many students in both UG3 and UG4. MSc students may take up to 20 credits at Level 10.]</i></p> <p><input type="checkbox"/> Level 11/Year 4 (also available in Year 5 and MSc). <i>[These courses are listed as options in both UG and MSc DPTs.]</i></p> <p><input type="checkbox"/> Level 11/PG (also available in Year 5). <i>[These courses are normally for MSc and UG5 students. They are not explicitly listed in UG4 DPTs, but UG4 students can take limited credits of them.]</i></p> <p><input type="checkbox"/> Level 11/PG (only). <i>[These courses are not available to UG4 or UG5 students. Examples: CDT courses; CPD courses.]</i></p> <p><b>Other options. Please provide justification if using:</b></p> <p><input type="checkbox"/> Level 9/Year 3 <i>[Deprecated except for compulsory UG3 courses. The course will not be available to other years.]</i></p> <p><input type="checkbox"/> Level 10/Year 4</p> <p><input type="checkbox"/> Other:</p>
<b>SCQF Credit Points</b>	<input checked="" type="checkbox"/> 10 <input type="checkbox"/> 20 <input type="checkbox"/> 40 <input type="checkbox"/> 60 <input type="checkbox"/> 80 <input type="checkbox"/> Other:
<b>Delivery Location</b>	<input checked="" type="checkbox"/> Campus <input type="checkbox"/> On-line Distance Learning
<b>Course Type</b>	<input checked="" type="checkbox"/> Standard (default) <input type="checkbox"/> Dissertation <input type="checkbox"/> Online Distance Learning <input type="checkbox"/> Placement <input type="checkbox"/> Student Led Individually Created Course <input type="checkbox"/> Year Abroad
<b>Marking Scheme</b>	<input checked="" type="checkbox"/> Standard (numerical) <input type="checkbox"/> Letter grade only <input type="checkbox"/> Pass/Fail <i>[Normally only for externally delivered courses]</i>

## Guidance for remaining sections:

**Before starting your proposal:** please contact the DDoLT (Curriculum) informally before starting to complete this form, with at least the following information:

- Tentative course title, level, year, and number of credits
- Who the target audience is, and why the course is needed.

The DDoLT (Curriculum) or delegate will schedule a meeting with you to discuss your plans and whether a full course proposal makes sense. If so, you will be provided with further instructions.

**Deadlines:** New courses must be approved by the December BoS meeting to ensure allocation of teaching staff for the following academic year. Since it may require considerable discussion and iteration to prepare the proposal, you should **contact the DDoLT (Curriculum) as early as possible, ideally in spring or summer**, and you should **plan on submitting your full proposal by November**.

**Submitting your proposal:** When your proposal is complete, please submit to [iss-bos@inf.ed.ac.uk](mailto:iss-bos@inf.ed.ac.uk).

### Colour coding and item-by-item guidance:

*Guidance is provided in italics for each item. Please also refer to the guidance for new course proposals at <http://www.inf.ed.ac.uk/student-services/committees/board-of-studies/course-proposal-guidelines>.*

*Examples of previous course proposal submissions are available on the past meetings page <http://web.inf.ed.ac.uk/infweb/admin/committees/bos/meetings-directory> but note that the proposal form was updated in Apr 2021.*

<b>Sections in gold</b> are for student view and are required before a course can be entered into DRPS.
<b>Sections in orange</b> are for School use but are still required for all courses (even those that have already been approved based on other documentation).
<b>Section in gray</b> are for consideration by the Board of Studies. They are normally required for all new course proposals but may be omitted in some cases, with permission (e.g., for invited proposals).

### Glossary of terms:

**(D)DoLT:** (Deputy) Director of Learning and Teaching.

**DRPS** ([The Degree Regulations and Programmes of Study](#)): Provides the University's official listing and descriptions of courses, degree programmes, and the regulations that govern them; updated annually in April. Course information in DRPS is considered a contract with students.

**DPT** (Degree Programme Table): Lays out the course requirements for each year of a degree. All UoE degrees have a DPT in the DRPS.

**Path:** A system that students use to help choose courses and view options in their DPT. The information feeds through from DRPS but has a more student-friendly interface (e.g., by highlighting courses that are not running or where the student hasn't satisfied prerequisites).

**SCQF** ([The Scottish Credit Qualifications Framework](#)): Lays out the requirements for courses at different levels and with different numbers of credits.

### 1. **Course overview and case for support**

Except as noted, all fields are required and will go into the DRPS (course catalogue) entry for students. **Important:** Text in DRPS is effectively a contract with students, so should not include details that are likely to change from year to year.

<p><b>Summary Description (for DRPS)</b>          This course further develops the introduction to testing in Informatics 2 - Software Engineering and Professional Practice. The course develops skills to select and apply a testing strategy and testing techniques that are appropriate to a particular development process, software system, or component. Class members will become capable users of test tools; will be able to assess the effectiveness of their testing activity; and will be able provide evidence to justify their evaluation. The course is assessed by portfolio demonstrating the achievement of the learning outcomes. This course is based on the IEEE Software Engineering 2004 Software Testing syllabus.</p>	
<p><b>Contribution to curriculum; target audience and expected demand; consultation (for BoS only)</b></p>	
<p>Why is this course needed and how does it relate to existing courses and degree programmes (including any prerequisite courses)?</p>	<p>This is a revision of the ST course to update the description to take account of curricular and scheduling changes and revise the assessment process.</p>
<p>What is the target audience, in terms of background and interests, and what is the expected demand (class size) for the course?  <i>State what your estimate is based on: e.g. by referring to projects in an area, sizes of similar courses, employer demand, etc. A survey of students may be requested once the main descriptor information is ready.</i></p>	<p>Third year students who have done the required courses for their degree programme. Masters students who are required to do IRR concurrently (I'm not sure why but this is in the current descriptor as a prerequisite)</p>
<p>Has this proposal been discussed with the DDoLT (Curriculum) or DoLT prior to BoS submission?</p>	<p><input checked="" type="checkbox"/> Yes  <input type="checkbox"/> No</p>
<p>Who else has been consulted?  <i>Proposals should typically be discussed with relevant colleagues, including the programme director (for MSc courses). Summarize their comments if needed.</i></p>	<p>I have consulted Stephen Gilmore and Michael Glienecke who are affected most directly by the proposed changes.</p>
<p><b>Course Description (for DRPS)</b>          Software testing is embedded in all software development processes and in recent years with the widespread adoption of DevOps and Continuous Integration, Delivery and Deployment, testing tools, automated testing, and the use of operational data in testing have become increasingly important. The course includes the following topics:</p> <ul style="list-style-type: none"> <li>• Testing techniques and principles: Defects vs. failures, equivalence classes, boundary testing.</li> <li>• Types of defects.</li> <li>• Black-box vs. Structural testing.</li> <li>• Testing strategies: Unit testing, integration testing, profiling, test driven development.</li> <li>• State based testing; configuration testing; compatibility testing; web site testing.</li> <li>• Alpha, beta, and acceptance testing.</li> <li>• Coverage criteria.</li> <li>• Test instrumentation and tools.</li> <li>• Developing test plans.</li> <li>• Managing the testing process: Development Lifecycles</li> </ul>	

- Problem reporting, tracking, and analysis.

Relevant QAA Computing Curriculum Sections: Software Engineering

Class members will take the development of tests for a small software project as their focus. For students taking the Informatics Large Practical course we recommend they take the software developed there as their focus. However, students are free to choose other software projects if they prefer. The goal is to gain experience of the full spectrum of testing techniques, test planning, testing process and demonstrate that experience on the chosen software project.

Class members will work in groups of 10 and will be encouraged to share experience by providing comment and reviews of others work. Weekly tutorials will be structured around different aspects of the course and the development of a portfolio describing the work done on the selected software project.

Each class member will develop an individual portfolio demonstrating they have achieved the learning outcomes of the course. This will use work on the testing of their chosen software project as evidence, augmented by appropriate other evidence. Acceptable kinds of evidence demonstrating achievement of the learning outcomes are diverse so part of the assessment is the design of the portfolio in advance of its construction. There are two or three “standard” portfolio designs but class members are encouraged to develop their own approaches that take account of their personal strengths and weaknesses. Portfolio designs will include specific assessment criteria.

Each week there will be a group meeting, around 1-2 hours of recorded material covering the lecture material in the course. Guest lectures given by a practitioner on their experience of testing in different contexts will illustrate the application of the concepts covered in the course in the development of a real-world product.

**Assessment Weightings (for DRPS)**

Written Exam \_\_\_\_0%  
 Practical Exam \_\_\_\_0% (*for courses with programming exams*)  
 Coursework \_\_\_\_100%

**Additional Information, Assessment (for DRPS)**

This is a 10-credit course, so it aims to develop skills in a range of techniques and demonstrate those skills as applied to the chosen software project. Course members will work in small tutorial groups discussing issues that arise for their chosen software project. Group members will provide comments and critiques of each other’s work. Individually, class members will develop a portfolio of evidence of attainment of the learning outcomes of the course. At an early stage in the course the design of each individual portfolio will be agreed to provide appropriate structure for the portfolio. Two or three “standard” portfolio designs are available and specimen portfolios are also available. Class members are encouraged to develop their own portfolio designs or modify standard designs to meet their learning needs.

**Learning Outcomes (MAXIMUM OF 5; for DRPS)**

On successful completion of this course, class members will be able to

1. Analyze requirements to determine appropriate testing strategies
2. Design and implement comprehensive test plans with instrumented code
3. Apply a wide variety of testing techniques and compute test coverage and yield according to a variety of criteria
4. Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise outcomes

5. Conduct reviews and inspections and design and implement automated testing processes

**Graduate Attributes, Personal & Professional Skills (for DRPS)**

*Unchanged.*

## 2. Additional information on course design and resourcing (for BoS only, except where noted)

### Breakdown of Learning and Teaching Activities (for DRPS)

Please fill in the number of timetabled hours per student for each type of activity. Do not include non-timetabled hours.

**A typical 10pt Informatics course has:**

- 18-20 lecture slots (2/wk), but only ~15h should be examinable lectures, with the rest used for guest lectures, revision sessions, assignment feedforward/feedback, etc. If unsure of plans, count these under 'lecture hours' but please explain tentative plans in the free text below.
- No more than 4-5 lab or tutorial hours. Please consider whether fewer can be used, e.g. by using some lecture hours for whole-class discussion/feedforward.

**A typical 20pt course has 30 lecture slots (3/wk) and no more than 8 lab/tutorial hours.**

Timetabled Hours	Type
18	Lecture Hours
10	Seminar/Tutorial Hours
0	Dissertation Project Supervision Hours
	Supervised Lab/Workshop/Studio Hours
2	Feedback/Feedforward hours
0	Summative assessment hours [Normally 2h if using an exam; otherwise 0]
0	Revision Session Hours

(Note for ISS: Remaining hours should be allocated to Directed and Undirected Learning Activities.)

### Use of timetabled activities (not to be included in DRPS)

- Each week will cover topics in sequence and the material on the topic will be available as a series of short videos.
- Most weeks there will be a guest lecture that is relevant to the current topic and is presented by an expert in the field or in a particular domain.
- Class members will work in groups of at most 10. Each group member will develop appropriate testing for their chosen software project. This will involve the development of a Continuous Integration process using a tool such as GitLab CI or Jenkins. Each week will focus on a different activity and students will work on reviewing each other's work. There will be weekly meetings of the groups with a tutor in attendance on alternate weeks. In non-tutor weeks students will work together following an activity sheet.
- Class members will construct a portfolio of evidence that they meet the learning outcomes of the course, in the first three weeks students will submit a design of their portfolio and will receive feedback on the acceptability of the proposed evidence. This may need revision to be acceptable.

### Summative assessment and time spent on assignments (not to be included in DRPS)

- Class members will work on developing testing for their chosen software project and on critiquing the work of others in their group. This will not be assessed but will be the primary source of evidence that they have achieved the learning outcomes of the course. This will be augmented with additional evidence.
- The assessment will be based on the individual portfolio. This will be structured according to the learning outcomes of the course and class members will provide evidence of attainment under each LO. Evidence will mostly be derived from the testing developed for their chosen

software project but can include other appropriate evidence. For example, contributions or critiques of the work of others, performance in tests provided in the course, oral presentations. The final deadline for the portfolio will be week 1 of semester 2 or revision week prior to the main exam diet depending on whether the course is presented in semester 1 or semester 2.

- The design of the portfolio should be complete by the end of week 3, this will cover the proposed contents of the portfolio, how the contents relate to the Learning Outcomes of the course, and, when each of the activities contributing to the portfolio will be completed. The design will be the subject of review and formative feedback this will provide a programme of work for the remainder of the course that is open to review.
- The use of a portfolio gives students experience in how to assemble a body of evidence to support an outcome (in this case the achievement of a learning outcome). It also stimulates reflection on how a student's activity is contributing to the objectives of the course.
- Portfolios will be assessed on the quality of the evidence of attainment of the LOs and that the evidence is the work of the portfolio owner. Clear assessment criteria will be developed as part of the development of the portfolio design.
- Example portfolios will be made available to students and there will be two "standard" portfolio plans that could be adopted by students if they choose.

#### **Tentative plans for feedback/formative assessment (not to be included in DRPS)**

The portfolio design will be formatively assessed and will plan out the planned learning and how it will be evidenced. Each meeting of the groups will consider progress against the portfolio design and will consider the quality of work and progress against the plan.

#### **Decolonisation and Inclusivity (not to be included in DRPS)**

**Content:** The literature on decolonisation of the Software Testing, Software Quality and Test Process is not extensive but there is a body of work on cross-cultural comparisons in software process that will be used to explore how choice and development of development process is shaped by the culture it is embedded in. We will draw on literature like this to provide a wider perspective on testing and development practice:

T. Wang and S. Wang, "A study on the influence of cultural differences on the behavior of software engineers/managers between Chinese and Scandinavians." 2019.

M. Ressin, C. Oyugi, J. Abdelnour-Nocera, D. Lee, and D. Panesar, "Exploring local cultural perspectives in user interface development in an Indian offshoring context: a view from the UK," in *International Conference on Human-Centred Software Engineering*, 2012, pp. 291–298.

T. Kano, "Soft Factors in Global ICT Sector Development: Studies with Bangladeshi and Rwandan ICT Workers," PhD Thesis, 2021.

E. Guveyi, M. S. Aktas, and O. Kalipsiz, "Human factor on software quality: a systematic literature review," in *International Conference on Computational Science and Its Applications*, 2020, pp. 918–930.

F. Fazli and E. A. C. Bittner, "Cultural influences on collaborative work in software engineering teams," 2017.

C. Consoli, P. Rocchi, and P. Spagnoletti, "An empirical study of offshore software development: The case of a ticketing application," *Journal of computing and information technology*, vol. 22, no. 4, pp. 267–275, 2014.

J. Abdelnour-Nocera *et al.*, "The centre for internationalization and usability: enabling culture-centred design for all," in *IFIP Conference on Human-Computer Interaction*, 2011, pp. 683–684.



Issues around the cultural shaping of testing and development process will be included in discussion and testing processes arising in any culture can provide the basis for the work of an individual in the class.

**Delivery:** The aim is to make the assessment as “democratic” as possible where students have choice on the form of evidence of attainment. The assessment design aims to enable class members to focus on their strengths and applying them to provide strong evidence of attaining the LOs for the course. This will also provide a means to adapt the assessment for students with disabilities. Since they will have extensive choice in how to construct their portfolio.

#### Anticipated Resource Requirements

If tutorials are needed, how many students per tutors? <i>(Please provide your desired number, and the maximum feasible number.)</i>	20 students per tutorial, i.e. two small groups of 10 students each. The tutorials will be structured to leverage the presence of two groups to promote learning and discussion. Tutorials will focus on different issues in software testing and different approaches taken depending on the student and chosen software project.
If labs are needed, how many students per demonstrator? <i>(Please provide your desired number, and the maximum feasible number.)</i>	<b>0</b>
Please estimate the number of hours required for marking, per student.	<b>1.5</b>
If any other teaching support resource will be requested in order to <b>develop</b> or <b>maintain</b> the course, please provide an estimate of that here.	
Do you anticipate any difficulty recruiting enough teaching support? <i>(For example if the course is very large or very specialized.)</i>	<b>No</b>
Does the course have any scaling limits due to available space or equipment?	<b>No</b>
If equipment is required, please state how it will be procured and maintained.	<b>None required</b>
Does the course have any external funding? <i>(Typically only for CPD courses)</i>	<b>No</b>
Does the course need any special arrangements such as quotas, agreements with other schools, or registration arrangements? Does it have any atypical characteristics that may affect finance or student registration? Please specify if so.	There is a link to the ILP course where students will be given the option to use their project in ILP as the software project they focus on in ST. I have discussed this with the CO of ILP. The ILP CO welcomes this approach.



### 3. Further information for BoS consideration

A full proposal for a new course must include examples of exercises and assessment. Please provide these below, along with publicity information.

#### **Course information and publicity**

The course is aimed at students on Sol courses, no additional publicity is necessary.

#### **Sample tutorial/lab sheet questions**

Each tutorial meeting of the group will have a group of questions/provocations that class members should consider before the meetings.

#### ***A first meeting tutorial sheet (this is an example)***

*In advance of your group meeting:*

- *You should read the overview of GitLab CI/CD (and watch the introductory video): <https://docs.gitlab.com/ee/ci/>*
- *You should have considered candidate software projects as the subject of your work on developing comprehensive testing for the project. In considering this you should consider what sorts of testing will provide good information on the quality of your candidate projects. A good candidate will need a range of testing approaches. Have a short description of at least one of the candidates ready to discuss.*
- *You should also consider your personal answers to the questions below in preparation for the group discussion.*
- *Decide with the group how you want to document your group meetings. You might want to make an audio recording and have it transcribed (word via o365 will do it for audio), or a video using some tool like stream (will make a transcript) or collaborate or have a minute taker whose primary task is to record group decisions and actions to be completed by specified individuals.*

#### ***Questions (Think about justifying your answers by identifying short quotations from the standard or otherwise)***

1. *Go round the group and find out how much experience of software development people in the group have (what sort of processes they have experienced and in what problem domain). Make a "directory" of experience of the group*
2. *Go round the group and find out what experience people have of software development, what tools they have used, and for what purpose. Add this to the directory of expertise.*
3. *Go round the group and have each person provide a brief description of a potential choice of software project. For each group member discuss briefly what testing seems most important for each potential choice of software project.*

#### **Sample assessment materials**

Assessment is by a single coursework submitted at the end of the course. The tutorials are devoted to developing the final submission and providing formative feedback on the development of the testing of their chosen project and on the construction of their portfolio.

Each class member works in a group of 10 students to develop and implement a testing approach for their chosen software project. This work will be summarised in a portfolio that links their work on testing to the Learning Outcomes of the course. The testing approach will involve a range of different methods, their application to the software project and an assessment of their effectiveness. Also, there will be consideration of the chosen testing process.

The testing approach, results and analysis are NOT evaluated directly. However, students are expected to refer to this within their portfolio. The portfolio is directed to demonstrating the student has achieved the learning outcomes specified in the course description:

- 1) **Analyze requirements to determine appropriate testing strategies**
  - a) Here we anticipate there will be justification of a choice of some testing approaches by linking to identified requirements.
  - b) There should be some consideration of how to measure key expected attributes of the system
  - c) There should be a coverage argument to say the planned approach is comprehensive
  - d) There should be some assessment of limitations
- 2) **Design and implement comprehensive test plans with instrumented code**
  - a) A plan should be developed that tracks the potential further evolution of the software project and proposed an approach to assessing the quality of the project.
  - b) The plan should demonstrably provide an approach that covers all the step identified in the previous step.
- 3) **Apply a wide variety of testing techniques and compute test coverage and yield according to a variety of criteria**
  - a) This should point to the implementation of the plan by choosing some appropriate tools and tests to implement the planned tests.
  - b) The adequacy of the tests should be assessed.
  - c) Some discussion of alternatives and the adequacy of chosen tools and process
- 4) **Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise outcomes**
  - a) This should point to the overall evaluation of the testing and how adequate it is.
  - b) This should demonstrate a clear understanding of the strengths and weaknesses of the student's work.
- 5) **Conduct reviews and inspections and design and implement automated testing processes**
  - a) This should point to reviews and inspections carried out during the development of the testing for the software project.
  - b) This should also point to automation considered.

#### 4. Additional Course Details for DRPS

Except where otherwise noted, these fields are required for entry into EUCLID and will be visible to students in the DRPS entry.

<b>Planned Academic Year of Delivery</b> <i>(The first year you anticipate the course running, e.g. AY 2019-20)</i>	AY 22-23
<b>Keywords</b> <i>Give a list of searchable keywords for the course.</i>	Unchanged
<b>Course Organiser</b> <i>(By default, the course proposer)</i>	Stuart Anderson
<b>Intended Delivery Period</b>	<input type="checkbox"/> Semester1 <input checked="" type="checkbox"/> Semester 2 (or semester 1 – I don't mind) <input type="checkbox"/> Full Year <input type="checkbox"/> Summer <input type="checkbox"/> Other (please specify):
<b>Timetable considerations/conflicts</b> <i>For School use. Please specify any constraints to be considered (e.g. overlap of popular combinations, other specialism courses, external courses etc). Include whether the semester delivery is constrained or could be flexible.</i>	
<b>Reading List/Learning Resources (for DRPS)</b> <i>You are encouraged to create resource lists using <a href="#">LEGANTO</a></i>	<p>I will use the existing list plus: [1]</p> <p>“The Fuzzing Book.” <a href="https://www.fuzzingbook.org/">https://www.fuzzingbook.org/</a> (accessed Feb. 08, 2022). [2]</p> <p>G. Fraser and J. M. Rojas, “Software Testing,” in <i>Handbook of Software Engineering</i>, S. Cha, R. N. Taylor, and K. Kang, Eds. Cham: Springer International Publishing, 2019, pp. 123–192. doi: <a href="https://doi.org/10.1007/978-3-030-00262-6_4">10.1007/978-3-030-00262-6_4</a>.</p>
<b>Feedback Information</b> <i>Provide a high-level description of how and what type of feedback will be provided to students, for inclusion in DRPS.</i>	<p>Students will receive weekly formative feedback on their work in the tutorials. They will also be required to develop a plan for their portfolio by the end of week three and will receive feedback on that. The summative feedback will evaluate the strength of the evidence provided by the student that they have attained the LOs of the course combined with an evaluation that they were responsible for the evidence.</p>

<p><b>Is this course available to visiting students?</b></p>	<p><input checked="" type="checkbox"/> Yes (default)  <input type="checkbox"/> No</p> <p><b>If no, please provide a justification here:</b></p>
<p><b>Required pre-requisite courses</b>  <i>Use sparingly: these are enforced in PATH and can only be waived by approval from the School's Curriculum Approval Officer. Note that cross-year required pre-requisites may prevent MSc students from registering; consider using recommended pre-requisites or "other requirements" instead.</i></p>	<p><u>Unchanged but if this is given in S1. IRR becomes a co-requisite for MSc students. OR With the permission of the Course Organiser.</u></p>
<p><b>Recommended pre-requisite courses</b></p>	<p><input checked="" type="checkbox"/> No  <input type="checkbox"/> Yes (please specify full course name(s) and code(s)):</p>
<p><b>Required co-requisite courses</b>  <i>Specify any courses that must be taken in parallel with the existing course. Note that this leads to a timetabling constraint that should be mentioned elsewhere in the proposal.</i></p>	<p><u>See above IRR is a co-requisite if this course is given in S1.</u></p>
<p><b>Prohibited Combinations</b>  <i>Specify any courses that may not be taken in combination with the proposed course].</i></p>	<p><input checked="" type="checkbox"/> No  <input type="checkbox"/> Yes (please specify full course name(s) and code(s)):</p>
<p><b>Other Requirements/Additional Information</b>  <i>This information is often used by MSc students and students from other Schools to see if they have appropriate background without having done our School's courses. So please avoid course titles, instead list specific knowledge and skills (such as mathematical concepts, programming ability or specific languages, etc).</i></p> <p><i>Also list any other constraints on registration, for example: "Only available to 4th Year Informatics students including those on joint degrees." or "This course is open to all Informatics students including those on joint degrees, and to students in the School of Mathematics. Other external students whose DPT does not list this course should seek permission from the course organiser."</i></p>	<p><input checked="" type="checkbox"/> No  <input type="checkbox"/> Yes (please specify):</p> <p>Students should have some basic understanding of software engineering and software lifecycle together with experience of programming (e.g. the material covered in the second year SEPP course in Informatics) .</p>

**Visiting Student Pre-requisites**

Same as "other requirements"

Different than "other requirements" (please specify):

**5. Placement in degree programme tables: for level 9-11 courses only (except EPCC)**

This section is for consideration by the Board of Studies and will be used later by ITO to determine where the course will be added to existing degree programme tables.

<p><b>Is this course restricted to students on a specific degree?</b>  <i>E.g., some courses are only available to students on a specific CDT or MSc.</i></p>	<p><input checked="" type="checkbox"/> No  <input type="checkbox"/> Yes (please specify and provide justification):</p>
<p><b>Is this course compulsory for students on any degree(s)?</b></p>	<p><input checked="" type="checkbox"/> No  <input type="checkbox"/> Yes (please specify and provide justification):</p>
<p><b>Any issues for part-time students?</b>  <i>Normally, part-time students have access to the same courses as full-time students on the equivalent degree. If you anticipate any problems with this, please specify here.</i></p>	<p>No</p>

**For optional courses:**

If this course is available but non-compulsory for students on various degrees (most courses), please fill in this section. The choices here determine where the course appears in degree programme tables (DPTs) and the 2-3 character tags are displayed in the Informatics sortable course list.

<p><b>Should this course be tagged as ‘ML’ (machine learning foundations and methods)?</b>  <i>Courses with the ML tag are typically very high-demand and most degrees limit the number of ML credits. If your course might appeal to a similar audience but draw off students from these large courses, please select 'no' and choose one of the tags below.</i></p>	<p><input checked="" type="checkbox"/> No  <input type="checkbox"/> Yes</p>
<p><b>If you chose ‘no’, please choose at least one of the following tags...</b>  <i>Ideally, select exactly one, unless there is a good argument for more than one. These three are used in various combinations for many of our degrees.</i></p>	<p><input checked="" type="checkbox"/> <b>FSS</b> (CS foundations, systems, and software)  <input type="checkbox"/> <b>AIA</b> (artificial intelligence applications and paradigms)  <input type="checkbox"/> <b>COG</b> (cognitive science: including HCI and NLP courses, but not most other AI courses. Please restrict to courses most relevant to natural cognition.)</p>
<p><b>...and also tick if any of the following tags or categories apply.</b>  <i>Do not tick any of these if you selected ‘ML’ already.</i></p>	<p><input checked="" type="checkbox"/> <b>SE</b> (software engineering: including courses that are highly relevant to SE degrees. All SE courses should also be FSS. This tag is mainly relevant for UG SE degrees.)  <input type="checkbox"/> Databases and data management systems (used for Data Science MSc and MSc(R))  <input type="checkbox"/> Unstructured data and applications (used for Data Science MSc and MSc(R))  <input checked="" type="checkbox"/> <b>Level 11 Security</b> courses (used for Security MSc)</p>

	<u>X</u> ATFC Optional courses (used for ATFC MSc)
<b>If you are not sure which tags are most appropriate or have other questions about this section, please note any comments/issues here.</b>	