# Online Learning for Informatics Courses

This is a proposal to pilot online learning in informatics courses. School of Informatics faces some significant challenges in delivering courses:

- **Scale**: some of our courses at level 8 and level 11 have more than 200 students on course and some have over 400. Managing courses at this scale is challenging using our traditional approaches.
- **Lack of Feedback**: many students feel they do not get adequate feedback to help them calibrate their learning against the learning outcomes of the course. Our traditional approaches using coursework provides relatively limited scope for feedback because 10 credit courses are limited to a single summative feedback opportunity. The scale of classes and complexity of our courseworks makes providing timely feedback challenging.
- **Engagement**: Difficulty to sustain student engagement given varied backgrounds and skill levels. Some classes suffer from low attendance.
- **Resourcing**: our traditional approach to hiring tutors and demonstrators does not scale well and we have difficulties with recruitment, and training of tutors and demonstrators.

Across the School there is considerable innovation in Teaching and Learning, but this is quite balkanised since staff typically only have time an energy for their own course and it is difficult to find effort to transfer expertise. This is particularly difficult in the case of courses with large classes where the administrative overheads restrict innovation. The “flipped classroom” initiatve driven by Bob Fisher has been the most systematic approach to innovation in teaching but we have limited experience of this for larger classes.

The pilot we propose tackles the scale and feedback issues directly and has a less direct but significant impact on resourcing. By providing good quality feedback to online activities the role of tutors and demonstrators will be more clearly defined and we believe this will ease some of our recruitment difficulties. It may also be the case that the development of online learning in the school will result in new roles that will be more attractive to our pool of potential tutors and demonstrators.

The remainder of this document motivates and specifies a request for modest additional resources in order to provide increased and diversified online content for the recently redesigned Informatics 1B course. We plan a similar approach for the distance version of the Introduction to Applied Machine Learning course but this is at an earlier stage of development and resource needs are less clear. This is predominantly human resource to allow the preparation of new materials and approaches that are described in the rest of this proposal. Once these pilots have been evaluated we will bring a larger and more detailed proposal to fund an initiative fully to establish a group to support online education within the School with the goal of providing effective online support to all our learners.

This document, following Paper 4.2 (outline of a new program for Advanced MSc) details a strategy for embedding online learning in existing courses. While Paper 4.2 specifically targeted administrative changes to the MSc program to meet the challenges of increasing cohort size, this document describes strategies aimed at combining online elements within our existing course structures to improve teaching and learning.

The goals of embedding online learning include:
- Promote student engagement in and out of the classroom,
- Increase mastery of skills
- Increase student classroom attendance,
- Adapt class material to students of varying knowledge backgrounds,
- Increase student skill mastery and  satisfaction.

Importantly, these goals should be achieved while minimizing the extra burden required by the course staff in order to facilitate and assess that learning.

Strategies can be separated into online technologies used inside and outside the physical classroom. For each of these strategies, we specify a scientific justification, the modus operandi, and some technological possibilities for implementation. I also inserted opportunities to involve AI in each setting.

**Interactive teaching**:
- What: Actively involve students in the learning process through interactive tests and conceptual questions in class.
- Where: during class.
- Why:
  - Yields immediate feedback about what students are learning through discussion with peers and/or instructors. Helps students gauge their own understanding.
  - promotes conceptual understanding (m and leads to better  learning outcomes (Freedman 2014).
  - Incentivizes participation for quiet or passive students by encouraging every student to think and do.
- How: Inclass polls (Piazza, Socrative, Google forms, tophat, mindmap applications (e.g. https://simplemind.eu/).
- Overhead for instructors: Preparing questions (already exists for many courses); reduced teaching time in class.
- AI: Adapt questions to skills and proven mastery  of individual students; team up groups to form a collaborative unit (see collaborative student learning)

**Collaborative student learning:**
- What: Students work with others towards a shared goal, being accountable to each other with only direction from the teacher.
- Where: during class, outside class.
- Why: Working in small groups provides learners with opportunities to articulate ideas and understandings, uncover assumptions and misconceptions, and negotiate with other students to reach consensus.

- How: Use small groups (3-4 students). Possibly, identify roles for the group members so everyone has a particular task to do and specific way to contribute. Groups can communicate online (outside of class) and face to face (during class).
- Overhead for instructors: The group's task should be purposeful, and should be structured in such a way that there is an obvious advantage to working as a team rather than individually.
- AI: Monitor group dynamics, Alerting instructors to "critical moments" in group activity. (requires clickstream of group behavior. )

**In-Context Course Forums:**

- What: Students anchor comments within course material during their learning.
- Where: outside class.
- How:web.hypothes.is/ (used in Iain Muray's class); Nota Bene (MIT)
- Why: Contexts provided by such forums allows students to consider all of the relevant discussion threads, improves the quality of the discussion and promotes student-to-student feedback (when compared to traditional temporal based forums).
- Overhead for instructors: Monitor comments, intervene when necessary.
- AI: Analysis of comments for cognitive engagement; intelligent visualization/summary; seeding new courses with best comments from past courses; recommendations on when to intervene.

**Peer grading:**

- What: Students grade each other's work.
- Why: to increase feeling of agency and empowerment by students (and decrease load from course staff). Peer assessment increases understanding and student achievement (can Zundert et al)

General online tools:

- badges for recognition of student work (e.g. https://badgr.com/)
- Online grading (Stack).

## Case study Inf1b (Introduction to OO with Java)

**Challenges**: Student numbers in the 300s; students vary greatly in background and expertise; low attendance; difficulty to sustain student engagement in class; low participation in class.

We suggest to do an active learning effort in class with close collaboration with the instructor: Engage students in classroom activities beyond listening, writing and memorizing. Use technology such as qualtrics or google forms. Students are prompted at key points in each lecture to a) express subjective belief about mastery of given skill (traffic light system); b) presented with 2 or 3 possible questions for different skill masteries; c) offered to solve the question at their reported mastery level. d) students are presented with correct answers, and aggregate results. e) students are surveyed during mid and final course with self evaluation surveys about master levels.

Requirements.

1. Data base of questions. Questions need to be non-trivial and appropriate for the class syllabus. Cover key concepts such as encapsulation, polymorphism, inheritance, containers etc.
2. Software for polling questions. Need to be able to configure topics, difficulty levels, and whether students solve individually or in groups.

**Resource:**

- Requirements can be fulfilled by a half-time student programmer over the summer. Estimation is 250 hours.
- Evaluation of approach (after implementation in Inf1B). Needs to include assessment of student and instructor satisfaction, performance, and comparison to past instances of the course.