Stellenbosch
UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

forward together
sonke siya phambili
saam vorentoe

MODULE FRAMEWORK: HONOURS PROJECT IN COMPUTER SCIENCE

COORDINATOR: W.H.K. BESTER

STELLENBOSCH UNIVERSITY, 2022

SUBJECT NUMBER	63444
MODULE CODE	711
CREDITS	32
MODULE NAME	Honours Project in Computer Science
SEMESTER	Both
ASSESSMENT MODE	Flexible
MODULE COORDINATOR	W.H.K. Bester, *whkbester@cs.sun.ac.za*
INTERNAL MODERATOR	W.D. Tucker, *btucker@sun.ac.za*
EXTERNAL MODERATOR	M.C. du Plessis, Nelson Mandela University

1. Each student SHALL have a project supervisor, and the supervisor's word SHALL be law, but…
2. The student SHALL arrange regular project meetings with the supervisor, subject to the supervisor's availability; the supervisor MAY choose to conduct group meetings.
3. There SHALL be five (and only five) deadlines, given in the table below.

WHEN	TYPE	DETAILS
End of term 1	F	Draft report, not exceeding seven pages, containing (i) an introduction, (ii) a literature survey [R] or requirements specification and basic software design description [SE], (iii) the experimental design [R] or plan for testing [SE], and (iv) bibliography.
End of term 2	S + F	Demo 1: report (as above) + demo.
Start of term 3	S	Demo 2.
End of term 3	F	Draft of final report.
End of term 4	S	Demo 3: final report + presentation + demo.

F = formative; R = research; S = summative; SE = software engineering.

   (a) All summative assessments SHALL include at least one additional reader or examiner.

   (b) The formative assessments SHALL be the only opportunities for which feedback by the supervisor on report writing is REQUIRED, but the supervisor MAY elect to require more regular draft report submissions.

   (c) For each formative assessment, written feedback SHOULD be provided to the student within two weeks of the submission; for Demo 1, this includes feedback by the additional examiner(s).

   (d) Feedback for formative assessments SHALL be either given (i) during a designated meeting with the student, (ii) as written remarks in the form of a conference or journal review, (iii) as written annotations on the submitted report, or (iv) some combination of the previous three items.

4. The final project mark SHALL be calculated according to the following weights for the summative assessments specified in Rule 3.

OPPORTUNITY	WEIGHT
Demo 1: report + demo	10%
Demo 2	20%
Demo 3: final report	30%
Demo 3: final demo	30%
Demo 3: final presentation	10%

5. All submissions SHALL be coordinated via SunLearn and the Computer Science GitLab server (see Rule 11).

6. The principal mode of communicating general project matters SHALL be via SunLearn, Microsoft Teams, and the CS Honours mailing list.

   (a) Students MUST ensure (i) that they have access to the CS Honours Project module on SunLearn, (ii) that their preferred e-mail addresses are registered for the CS Honours mailing list, and (iii) that they have access to the appropriate CS Honours Project teams on Microsoft Teams, as communicated via SunLearn announcement or e-mail, and *before* any online session.

   (b) For urgent announcements, students MUST ensure that their SunLearn profiles are set up to forward announcements to e-mail.

   (c) If a student misses important or urgent announcements, incorrect software or device setup SHALL NOT be accepted as an excuse.

7. A student may lodge an application for a leave of absence from any summative assessment given in Rule 4 with the honours project coordinator. Such an application MUST follow the schedule and will be judged according to the rules set out in §11 of the chapter titled "Admission and Registration" of the Stellenbosch University Calendar (Part 1).

   In what follows, "granted a leave of absence" means that a leave of absence was granted either by the Registrar (according to §11, "Admission and Registration", Calendar Part 1) or the Computer Science Division, whereas "leave without absence" means that either an application for a leave of absence was denied by the Registrar or the Computer Science Division, or that a student was absent and did not apply for a leave of absence. "Demo 4" refers to an additional assessment opportunity offered under the conditions set out below.

   (a) If a student is granted a leave of absence for Demo 1 or 2, then the assessment opportunity involved SHALL be rescheduled without penalty.

   (b) If a student is absent without leave from Demo 1 or 2, then a mark SHALL NOT be awarded for the assessment involved, and as such, implies a mark of zero for this assessment.

   (c) If a student is granted a leave of absence for Demo 3, then access to Demo 4 SHALL be granted automatically, without penalty.

   (d) If a student is absent without leave from Demo 3, then access to Demo 4 SHALL be granted automatically, but the maximum final project mark SHALL then be 50.

   (e) If after participating in Demo 3 a student has a final project mark of less than 50 but at least 40, then access to Demo 4 SHALL be granted automatically, where the marks of Demo 4 SHALL replace the marks of Demo 3, and the maximum final project mark after Demo 4 SHALL be 50.

(f) Demo 4 SHALL consist of a project report, demo, and presentation at the same weight distribution as Demo 3 (for a total of 70%), and SHALL be scheduled in the month before the start of the *next* academic year; therefore, students who have access to Demo 4 will not be able to graduate in December of the current academic year.

(g) If a student who was allowed access to Demo 4 does not participate in this opportunity, the marks for Demo 3 (if any) SHALL stand.

(h) There SHALL NOT be any further project assessment after Demo 4.

8. If a student fails the project and is allowed to reregister for the next academic year, then continuing the same project topic or working under the same supervisor SHALL NOT be allowed.

9. There SHALL be one LaTeX class for the project report, which MUST be used by all students, regardless of research or software engineering focus.

   (a) The page limit SHALL be 20 pages, excluding title page, table of contents, and list of references.

   (b) The supervisor MAY instruct the student to include material in one or more addenda, which does not count towards the page limit.

   (c) The font size or other layout parameters MUST NOT be changed. Any attempts to influence the default text density SHALL be penalised.

   (d) The student MAY use any package on *The ACM Publishing System (TAPS) List of Accepted LaTeX Packages* (`https://www.acm.org/publications/taps/accepted-latex-packages`), except that such packages MUST NOT be used in contravention of Rule 9(c).

   (e) For the inclusion of graphics, (scalable) vector formats MUST be preferred over bitmapped formats. For example, plots, charts, and UML diagrams SHOULD be rendered via (export to) SVG, PDF, T*i*kZ, or PGF, but photographs SHOULD be included as JPEG data.

10. Each final report submission MUST contain a clear indication of lines-of-code (LOC) written by the student.

    (a) The supervisor MAY instruct the student to contextualise LOC, for example, by remarking on integration with an existing code base.

    (b) Where applicable, the student MUST account separately for autogenerated code and of how much of the functionality is provided by external libraries or an existing code base (or both).

11. Each student MUST maintain two Git repositories on the Computer Science GitLab server: (i) one for the experimental software artefacts [R] or application code [SE], and (ii) another for the final report.

    (a) The repositories SHALL be set up at the beginning of the academic year by the technical officer, and MUST follow a standard URL naming convention that includes the project identifier.

    (b) Each code repository SHOULD use containerisation for a standard Unix system, so that results can be verified [R], or so that the application and test cases can be run for evaluation [SE]. Where this is thought not possible,

        i. the student MUST apply to the honours project coordinator for an exemption by the end of the first semester, and

        ii. in the application, reasons for *not* complying MUST be clearly given.

    (c) Each code repository MUST contain a `README` file with instructions on (i) how to start up the environment, including automatic resolution of any dependencies, (ii) how to run the application, and (iii) how to run experiments or tests.

(d) All academic staff SHALL have *read* access to all project repositories.

(e) Where privileged data is involved, students MUST in consultation with supervisors arrange for this data to be housed separately in a suitable way so that the project code can still be assessed.

(f) Unless approved by the honours project coordinator, a project repository MUST NOT be publicly available through any external service.

12. Each student SHOULD have a "dry run" for the presentation with the supervisor.

13. All presentation slides MUST include a slide number on each (logical) slide, and SHOULD include the total number of slides. Not having slide numbers SHALL be penalised (Demo 3 presentation mark).