UNIVERSITY OF KENT – CODE OF PRACTICE FOR QUALITY ASSURANCE

MODULE SPECIFICATION

1 **The title of the module**

CO538 Concurrency Design and Practice

2 **The Department which will be responsible for management of the module**

Computer Science

3 **The Start Date of the Module**

September 2009

4 **The cohort of students (onwards) to which the module will be applicable**

2010/2011

5 **The number of students expected to take the module**

40

6 **Modules to be withdrawn on the introduction of this proposed module and consultation with other relevant Departments and Faculties regarding the withdrawal**

None

7 **The level of the module (eg Certificate [C], Intermediate [I], Honours [H] or Postgraduate [M])**

Intermediate (I)

8 **The number of credits which the module represents**

15

9 **Which term(s) the module is to be taught in (or other teaching pattern)**

Autumn

10 **Prerequisite and co-requisite modules**

CO320 is a prerequisite. CO523 is an *alternative* corequisite to CO320.

11 **The programmes of study to which the module contributes**

BSc Computer Science (and related programmes)

BSc Mathematics with Computer Science

BSc Applied Computing (Joint Honours)

plus Year in Industry variants

12 **The intended subject specific learning outcomes and, as appropriate, their relationship to programme learning outcomes**

On successful completion of this module, students will:

- have an understanding of concurrent programming using a communicating process model (*process oriented design*) and how this differs from *object oriented design* [A2, A3];
- be able to design and implement communicating process networks in both diagram and code forms;
- be able to apply knowledge of concurrency when designing and implementing process networks [A5, C1];
- understand the dangers of deadlock, livelock and starvation inherent in concurrent programming;
- have an understanding of the types of application for which concurrency is important [A1].

13 **The intended generic learning outcomes and, as appropriate, their relationship to programme learning outcomes**

By the end of the module students should:

- understand trade-offs involved in design choices in concurrent systems; [B1]
- be able to evaluate and implement the design of simple concurrent software systems. [B5]
- understand and be able to use computational thinking for the solution of problems. [B7]
- be able to make effective use of IT facilities [D3];
- be able to manage their own learning and time [D5].

14 **A synopsis of the curriculum**

The CSP model of concurrency is the core of this module, primarily presented through the occam-pi multiprocessing language and additionally using the JCSP library for Java. The following topics will be covered:

- Parallel programming: processes, channels, alternation, barriers, CREW principle, types, variables and computation.
- Building concurrent systems: components, layered networks and modularity.
- Semantic aspects of concurrent programming: security-issues, race-hazards, deadlock, livelock, starvation and non-determinism.
- Design methodologies for deadlock, livelock and starvation-free systems.
- Performance of concurrent systems (including the efficient exploitation of multicore processors).
- Applications of concurrency for: robotics, embedded-systems, networking, complex system modelling, graphics and hardware design.
- Implementation of concurrent systems in natively sequential languages, comparing the use of specialised concurrency models (e.g. CSP) with a standard threads-and-locks model.

15 **Indicative Reading List**

Course notes, including copies of the lecture-slides and additional material, will be available in printed booklet form.

Other reading:

A Tutorial Introduction to Occam2 Programming. R. Pountain and D. May, BSP Professional Books (ISBN: 0-642-01847-X); 1987.

Occam2 (including occam2.1). J. Galletly, UCL Press (ISBN: 1-85728-362-7); 1996.

Parallel Processing with Communicating Process Architectures. I. East, UCL Press (ISBN: 1-85728-239-6); 1995.

16 **Learning and Teaching Methods, including the nature and number of contact hours and the total study hours which will be expected of students, and how these relate to achievement of the intended learning outcomes**

All learning outcomes will be achieved through a combination of lectures, classes and private study, supported by reading guides and web-based material. Further assistance will be available electronically via newsgroups and the web. Achievement of the learning outcomes will additionally be facilitated by formative coursework assignments, supported through the same means. This module comprises of 150 hours of study, broken down approximately as follows:

22 hours of lectures

14 hours of classes (4 terminal sessions and 10 seminars). Initially, one terminal class and seminar per week – then one seminar per week

40 hours spent on coursework

74 hours of private study (following up taught material and including exam revision)

Seminars (of 10-15 students) will support the lectures by reinforcing key ideas in an environment where students can discuss both lecture and coursework material. Terminal sessions will provide an opportunity for students to apply knowledge and demonstrate understanding in the context of specific case-studies.

17 **Assessment methods and how these relate to testing achievement of the intended learning outcomes**

All learning outcomes are assessed by coursework and an unseen written examination. The weightings are as follows:

- 50% written examination
- 50% coursework

Coursework is used to assess the *practical* learning outcomes – specifically: use of an appropriate concurrency IDE for the edit-compile-run-debug cycle (D3), ability to understand and apply a small library of simple processes in the construction of networks with richer behaviour (A2, A3, A5, C1), ability to specify, design and manage the development of larger systems with high levels of concurrency through an open-ended exercise (A1-A3, A5, C1, B1, B5, B7, D5). A written class test

(taken in exam conditions) will also be used to assess *theoretical* aspects of the aforementioned learning outcomes (A1-A3, A5, B5, B7, C1).

The written examination will assess whether the Learning Outcomes have been mastered sufficiently so that (a) they can be applied to the solution of concurrency problems in previously unseen contexts and (b) the student can reflect on them in a structured and mature manner. All the subject specific outcomes will be assessed: specifically: understanding of the concurrency primitives within the CSP/occam-pi model, application of these primitives both in small and large systems, theoretical understanding of concurrency security issues (e.g. race hazards, deadlock, livelock and starvation) and how to avoid them through good design, understanding and exploitation of non-determinism, comparison between process orientation and object orientation, applications of concurrency (e.g. for the efficient use of mulitcore processors).

Classes are not assessed, but do contribute to the learning outcomes assessed by coursework and examination.

18 **Implications for learning resources, including staff, library, IT and space**

This module will require staff to deliver the module, through teaching, supervision and marking. The library should have copies of the additional reading material. This module also requires the use of appropriate free software, available on public PCs.

19 **SENDA Statement**

The School recognises and has embedded the expectations of current disability equality legislation, and supports students with a declared disability or special educational need in its teaching. Within this module we will make reasonable adjustments wherever necessary, including additional or substitute materials, teaching modes or assessment methods for students who have declared and discussed their learning support needs. Arrangements for students with declared disabilities will be made on an individual basis, in consultation with the University's disability/dyslexia support service, and specialist support will be provided where needed

Statement by the Director of Learning and Teaching: "I confirm I have been consulted on the above module proposal and have given advice on the correct procedures and required content of module proposals"

....

Director of Learning and Teaching Date

Statement by the Head of Department: "I confirm that the Department has approved the introduction of the module and will be responsible for its resourcing"

....

Head of Department Date

Revised 25 August 2010