

PROPOSAL TO INTRODUCE A NEW COURSE

1. COURSE DETAILS

1.1 **Course ID** COMP9171

1.2 **Course name – Long** Object-Oriented Programming

1.3 **Course name – Abbreviated** Object-Oriented Programming

1.4 **Course Authority** Assoc. Professor Jingling Xue **ext/email** 54889/jxue@cse.unsw.edu.au

1.5 **Organisational Unit responsible for course**

School: School of Computer Science and Engineering

Faculty: Engineering

Academic Group Code (Faculty): ENG

Academic Organisation Code (Owner): COMPSC

1.6 **Justification of Proposal**

COMP4001 Object-Oriented (OO) Software Development, consists of two modules, OO programming (with an emphasis on C++) and OO design. There are usually about seven lectures for each module, two individual programming assignments in C++ and one substantial four-people group project on OO design.

As an elective, this course has had decent enrolment numbers in the past: 132 (05x1), 144 (05s1), 75 (06x1) and 91 (06s1). The enrolment numbers for postgraduate students are 19 (05x1), 30 (05s1), 25 (05x1) and 24 (0s61). The popularity of this course can be attributed partly to the increasing importance of OO technology to industry and partly to the OO programming and design skills that students have acquired through completing this course.

However, there is simply too much to cover both modules in reasonable depth in one course. In addition, two seemingly independent modules are closely coupled. On one hand, OO design is better understood in concrete OO languages, requiring students to be reasonably proficient in C++. On the other hand, OO programs are better developed if students have already had some basic understandings about OO design principles. It is believed these two problems combined may be an important source of dissatisfaction for some students in the past despite that the course has been popular (in terms of enrolment numbers).

The aim of this proposal is to make the OO programming module available in a new course, thereby effectively splitting COMP4001 into two separate ones: COMP9171 Object-Oriented Programming and COMP4001 Object-Oriented Software Development.

We are also proposing COMP3171 Object-Oriented Programming for undergraduates. This course would be taught concurrently with COMP9171, and the proposed handbook entry reflects this. There is a separate proposal for COMP3171.

1.7 **Consultation Process**

The proposal promises to solve the two main problems outlined in Section 1.6. It is introduced based on my own experience in teaching this course, my interactions with students, consultation with two other lecturers, Guy Tsafnat (who taught it recently) and John Potter (who designed the current

implementation) and on the CSE Course Experience Survey Data for this course collected over the past few years.

In addition, Bill Wilson, John Shepherd and John Plaice were consulted. John Plaice ``heartily supports the proposal” and ``would be very willing to assist in developing the course”.

1.8	Units of credit (UOC)	Session/s offered	Hours Per Week
	6 UoC	S2	3 hours

1.9 **Pre-requisites:** COMP9024 or enrolment in program 8684
Co-requisites: none
Exclusions: none

1.10 **Proposed Entry in the Faculty Handbook** (including course description)

This course covers the fundamentals and advanced techniques of object-oriented programming in C++.

Object-oriented inheritance techniques. Advanced techniques with functions. Memory and resource management. Namespaces; Run time type information; Templates and generic programming. C++ Template metaprogramming. The Standard Template Library (e.g., algorithms, containers and iterators). Input/output with C++ iostreams library. Exception handling. C++ and Efficiency Issues. Effective C++ design guidelines.

This course may be taught concurrently with COMP3171.

1.11 **Is this course replacing an existing course?**

Partly replaces COMP4001 Object-Oriented Software Development

1.12 **Postgraduate**

1.13 **Elective**

1.14 **Program stage**

Stage 2 in coursework programs that require more than one year
Stage 1 in programs that require at most one year.

1.15 **Program/s in which course is be available**

8682 MCompIT
8684 MIT
5432 Graduate Diploma in Computing & Information Technology
7344 Graduate Certificate in Advanced Computing
1650 PhD
2665 ME
2765 MSc

1.16 **Proposed teaching methods and assessment practices**

Teaching: lectures, tutorials and projects.

Assessment: programming assignments and written exam.

1.17 Assessment grades to be used

Full range of UNSW grades (i.e. FL,PS,CR,DN,HD,etc.).

1.18 Mode of delivery

Internal	X
External	
Other (specify)	

1.18.1 Multi-mode Delivery Guidelines

Not applicable

1.19 Information Technology Requirements for students

Standard resources available in the school.

1.20 Textbooks

Text: Stanley B. Lippman, Josée Lajoie, Barbara E. Moo: C++ Primer (4th Edition). Addison-Wesley, 2005. ISBN 0-20172-148-1.

Recommended:

1. Bjarne Stroustrup: The C++ Programming Language (Special Edition). Addison Wesley. 2000. ISBN 0-201-70073-5.
2. Nicolai M. Josuttis, Nicolai M. Josuttis, David Vandevorode: C++ Templates: The Complete Guide, Addison Wesley, 2002. ISBN: 0-201-73484-2.
3. Nicolai M. Josuttis: The C++ Standard Library: A Tutorial and Reference. Addison-Wesley, 1999. ISBN 0-201-37926-0.
4. Herb Sutter: Exceptional C++ Style. Addison-Wesley, 2005. ISBN: 0-201-76042-8.
5. Scott Meyers: Effective C++: 55 Specific Ways to Improve Your Programs and Designs, 3rd Edition. Addison-Wesley. ISBN: 0-321-33487-6.
6. Stanley Lippman: Inside the C++ Object Model. Addison-Wesley. 1996. ISBN 0-201-83454-5.
7. David Abrahams, Aleksey Gurtovoy: C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond (C++ in Depth Series). Addison-Wesley, ISBN 0-321-22725-5.

1.21 Industrial experience component

Not applicable.

1.22 Parallel Teaching Requirement

This course may be taught concurrently with COMP3171 Object-Oriented Programming. It is an elective course in the MCompIT, MIT and associated graduate certificates and diplomas, and may be taken by research students as part of their coursework requirement.

Both COMP9171 and COMP3171 are taught concurrently under Recommendation 3 of the Academic Board Policy on Parallel Teaching (Resolution AB04/106).

Lectures and some assessment tasks are delivered concurrently.

a) The handbook course description contains a statement that advises students that parallel

teaching of postgraduate and undergraduate course is occurring.

- b) Postgraduate students are enrolled under a postgraduate course code.
- c) Postgraduate students are required to complete an additional assignment.
- d) Postgraduate students and undergraduate students take different final examinations.

2. RESOURCE STATEMENT

2.1 Enrolments

Estimated or proposed enrolments for the next three years.

2007: 30
2008: 40
2009: 50

2.2 Resource Requirements

Staffing Requirements:

Hours per week

3	Full-time Academic Staff
1 hour per 20 students	Part-time Teaching Staff
0	General Staff

Field Costs:	N/A
Studio/Laboratory Requirements:	N/A
Materials Requirements:	N/A
Equipment Costs:	N/A
Computing Requirements:	Standard for CSE courses, already available.
Library Requirements:	2 copies of text in the open section and 3 in the reserved section. 2 copies of each reference book in the library.
Capital Funds Requirements:	N/A

2.3 Servicing Implications: N/A

2.4 Teaching Arrangements:

- (i) Will other units contribute on a regular basis to the teaching of this course? NO

2.5 Alternative Delivery Arrangements: N/A

2.6 Details of Tuition Fees:

Standard fees for courses in the Faculty of Engineering.

3. AUTHORISATION

3.1 University Librarian's Endorsement

Note: *this section of the Proposal must be signed by a Library representative, stating:*

I have examined the Library needs related to the above proposal and certify that existing Library holdings, staffing, services and accommodation are adequate / inadequate (delete one) to cover the demands that are inherent in it.

Appropriate arrangements for the use of digitised material to support this course have been made by the Course Authority with the University Librarian.

Further Comments:

University Librarian
/ /2006

3.2 Head of School's Approval

Note: *this section of the Proposal must be signed by the Head of School, stating:*

I have examined the resource implications of the above proposal in regard to staff, space, materials, equipment, capital funds, and computing, and certify that the School can cover the demands that are inherent in it.

Further Comments:

Head of School
/ /2006

3.3 Dean's Approval

Note: *this section of the Proposal must be signed by the Dean, stating:*

I have examined the resource implications of the above proposal in regard to staff, space, materials, equipment, capital funds, and computing, and certify that:

(Tick whichever is applicable)

3.3.1 (i) the proposal involves no additional resources. (A statement from the Head of School explaining how this can be achieved must be provided); or

- (ii) the proposal involves additional resources and it is proposed to redeploy existing resources within the faculty. (A statement from the Head of School explaining how this will be achieved must be provided); or
- (iii) the proposal involves additional resources to be obtained as set out below; or
- (iv) the additional resources essential to bring the proposal into effect cannot be found within resources available to the faculty.

3.3.2 **Fees** (delete if not applicable):

- a fee will not be charged for this program (other than HECS)
- a fee will be charged for this program for local fee-paying students
- a fee will be charged for international students

If a fee is to be charged the Dean certifies as follows:

I have ensured that the Vice-Chancellor has been advised of the proposed fee arrangements, and note that approval of fee arrangements is needed before the new program can be implemented.

3.3.3 the proposal conforms to the University's commitment to Equal Opportunity in Education.

Statement from Head of School on Source of Additional Resources and/or Further Comments:
--

Dean
/ /2006