FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

SCHOOL OF INFORMATION TECHNOLOGY

COURSE OUTLINE

3202INT Programming with Procedural Languages

1.0 Identifying Information

Course catalogue no :	3202INT
Course title :	Programming with Procedural Languages.
Field of Education Code	
Program/s	Bachelor of Information Technology (Industry)
School :	Information and Communication Technology
Faculty :	Engineering and Information Technology
Status of Course within program/s or academic plan/s	Core
Credit point value	10CP
Prerequisites :	
Year and semester :	Semester 2, 2005
Course convenor	Greg Cranitch
Teaching team members :	Dr Philippe Martin
Date course outline was last modified	26th July, 2005

2.0 Objectives

2.1 *Purpose of the course.*

The purpose of the course is to provide students with

- the opportunity to understand the structure of COBOL programs
- algorithms for some common COBOL file processing problems
- strategies or conventions for designing clearer and more maintainable programs
- strategies for testing and debugging COBOL programs
- access to COBOL resources that may be useful in their programming activities.

Upon successful completion of this course, students should be able to:

- develop and debug a range of COBOL programs
- develop conventions and test strategies for COBOL programs
- critically review current issues in COBOL development.

3.0 Interrelationship of the Course with other Courses and the Program

This is a core course.

4.0 Brief Description

This course directly addresses the development of commercial applications in COBOL. Common syntax is presented. Strategies are presented to help with the development, debugging and testing of COBOL programs. Some algorithms for common COBOL problems are presented. Assessment is by assignment and model development.

5.0 Content

Week 1	Introduction; From Coding Form to Computer	Chapters 1, 2
	Getting Started	Appendix A
Week 2	A Methodology for Program Development	Chapter 3
	Identification, Environment, and Data Divisions	Chapter 4
	The Procedure Division	Chapter 5
	(the requirements for the first COBOL program are given)	
Week 3	Debugging; Editing and Coding Standards	Chapters 6, 7
	Data Validation	Chapter 8
Week 4	Introduction to the Marist mainframe system	
	More about the Procedure Division	Chapters 9
Week 5	Introduction to Tables; Table Lookups	Chapters 11, 12
Week 6	Multilevel Tables; Sorting	Chapters 13, 14
	(the first COBOL program is due)	
Week 7	Control Breaks; Subprograms	Chapters 15, 16
	(the requirements for the major COBOL project are given)	
Week 8	Sequential File Maintenance	Chapter 17
Week 9	Indexed Files	Chapter 18
Week 10	*Free week (the major COBOL project is due week 12)*	
Week 11	Usage of DB2 from Cobol	
Week 12 & 13	JCL for larger jobs within COBOL	

(the major COBOL project is due)

(Note: the final examination will not include questions or exercises on Chapter 20 and the appendices)

6.0 Generic Skills Development

Skill Attributes	Level taught / practiced
Teamwork	Introductory
Problem solving & decision making	Advanced
Analysis & critical evaluation	Advanced
Written communication	Introductory
Self-management skills	Introductory
Inter-personal skills	
Oral communication	Intermediate
Information skills	
Adaptability & learning skills	Introductory

7.0 Flexible Learning

This course will be taught in Mode B – Web Dependant. Lecture notes, tutorial questions, lab guides and assessment item material will be available on the Internet. There are no face-to-face classes.

8.0 Rationale for Content

Many major corporations and large government departments have a significant investment in COBOL programs. These need to be maintained and enhanced. This course provides the fundamental knowledge that graduates would require to develop, test, debug and maintain COBOL programs.

9.0 Organization and Teaching Methods

This course is taught over the internet. Students are expected to work through the weekly materials. The assessment allows the students to design, develop, debug and test COBOL programs. The final examination allows students to demonstrate their ability under examination conditions. Students will be provided with the opportunity to develop and test their COBOL programs on a realistic mainframe environment.

10.0 Rationale for Teaching Methods

Lecture: Students are expected to familiarize themselves with content by reading the text and lecture notes. The weekly materials are an efficient vehicle for a brief review of that content, a discussion on it's place in the course with regards to objectives and assessment items, and a discussion on the relevance of the material to the student and it's real world application. Students may ask questions to seek further clarification.

Labs: Students must have an opportunity to work with and learn the software used to create assessment items, before those assessment items are due. This will give students an opportunity to explore for themselves and create questions in their own mind. Instructional material and assistance via the internet will be provided to answer those questions.

Griffith Uni and Global Online Learning will maintain a relationship with the Marist system to provide access for the students to a mainframe environment. Practical illustrations would be possible in this environment.

11.0 Assessment

There are four assessment items as detailed below.

NO.	*DESCRIPTION*	*WEIGHTING%*
1.	*First COBOL program due end of week 6*	*10*
2.	*Major COBOL project due end of week 12*	*25*
3.	*5 exercises due weeks 3, 5, 7, 9, 13*	*15*
4.	*Final examination during exam period*	*50*

All assignments are due at 5pm on the Friday of the week. All assignments are to be delivered via the Drop box on the Learning@Griffith course site.

To gain a grade of pass or better students must obtain at least 40% in each of the four assessment items and gain a pass mark overall.

12.0 Rationale for Assessment

The assignments allow the students to demonstrate their practical abilities in the design, development, testing and debugging of COBOL programs.

The 5 fortnightly exercises allow students to demonstrate their ability in a range of situations that cannot be easily demonstrated in the major programming assignments. These exercises will help with the programming assignments.

The examination allows students to demonstrate their ability under examination situations without external assistance.

13.0 Texts and Supporting Materials

Textbook:

Grauer, R.T., Villar, C. V., and A.R Buss. (2000) COBOL: From Micro to Mainframe, 3rd Ed. Prentice Hall. (Edition with the Fujitsu COBOL 4.0 Compiler)

Supporting materials from the Marist system.

Other recommended references: 1) www.prenhall.com/, 2) Murach's "Mainframe Cobol" (full details: Murach's "Mainframe COBOL" by Mike Murach, Anne Prince and Raul Menendez, 22 chapters, 687 pages, 310 illustrations, ISBN 1-890774-24-3).

14.0 Scope of Course Evaluation

This course will be evaluated in accordance with the current School of Information Technology procedures for course evaluation.

15.0 Administration

15.1. To be eligible to pass the course, students are required to complete ALL forms of assessment and must demonstrate a reasonable degree of competence in the required course objectives as examined in each form of assessment.

15.2. Students must obtain at least 40% in the each piece of assessment and 50% overall to gain a passing grade.

15.3 Non-submission of a piece of assessment will incur a fail grade for the course.

15.4 Students may work together in researching their assignments but final submission must reflect the work and original contribution of each individual student.

Any dishonest assignments will be dealt with under the rules applying in "The Process of Assessment, Grading and Dissemination of Results" and Statute 8.2 - Student Good Order as defined in the University Handbook.

Dishonest assignment includes:

- deliberate copying or attempting to copy the work of other students;
- use of or attempting to use information prohibited from use in that form of assessment;
- submitting the work or another as your own; or
- plagiarism (i.e. taking and using as your own the thoughts and writings of another with the intent to claim the work as your own).

15.5 Full and detailed acknowledgment (e.g. notation, and/or bibliography) must be provided if contributions are drawn from the literature in preparation or reports and assignments.

15.6 Documents for assessments must be created with a text editor.

15.7 Students must be able to produce a copy of all work submitted if so requested. Copies should be retained until after the release of final results for the course.

15.8 Assignment submissions must contain only files relating to that assignment. Submissions containing irrelevant files and / or viruses may NOT be assessed. Files must be named as advised by the course convenor. Files must have accurate date and time labels attached to them.

15.9 Assignment MUST be submitted by the due date and time. Extensions may be granted in exceptional circumstances by “Application for Extension” and MUST be made BEFORE the due date. Extension Application Forms are available from the Administration Office of the Faculty. Before an extension will be granted, a review of the work completed to date MUST be undertaken with the course convenor.

15.10 An assessment item submitted after the due date, without an approved extension, will be penalized. The standard penalty is the reduction of the mark allocated to the assessment item by 10% of the maximum mark applicable for the assessment item, for each day or part day that the item is late. Weekends count as one day in determining the penalty. Assessment items submitted more than five days after the due date are awarded zero marks awarded zero marks.

16.0 Course Communications

Important announcements relevant to the course will be placed on the blackboard notice-board for the course.