



Wollongong College Australia

A College of the University of Wollongong

Australia

Wollongong

ITC Education Ltd trading as
Wollongong College Australia
CRICOS 02723D
ABN 14105312329

Diploma in Information Technology

(CRICOS course codes: 057233A, 057234M)

Subject Outline Summer 2009/10

WUCT114 Procedural Programming

© Wollongong College Australia. All rights reserved 2009.
WCA-WUCT114-S0/4

Procedural Programming

Subject Description

This subject is designed as an introductory course in computer programming and is suitable for students who have no prior knowledge of programming languages or who have little (or no) experience in designing, developing and implementing programs in the C++ programming language.

The lectures introduce students to computer programming by explaining the fundamental concepts involved in programming computers. Programming is treated as total creative process by which programs are written for the purpose of instructing computers to carry out a variety of tasks.

To develop skills in the design, development and implementation of programs, students will be given a number of assignments and will be shown how to develop a programming solution to each specified problem by performing a number of logical steps. These assignments will provide practical experience in writing, debugging and testing programs on the C++ Integrated Development Environment.

The subject content includes computer algorithms, program design, implementation and documentation. Elements of "good programming style" will be treated as a significant part of the course.

Subject structure

Procedural Programming is delivered in a face-to-face format of 7 (seven) hours per week. The subject consists of 1 x 3 hour lecture, 1 x 1 hour tutorial and one 3-hour practical session each week. The session is of 14 week's duration with face-to-face classes scheduled for the first 12 weeks and a study/examination period in Weeks 13 and 14.

Students may be required to access computer laboratories outside normal laboratory class time in order to complete assignments. Furthermore, students will be required to undertake supplementary reading from relevant textbooks. Students will need to bring their own storage device (such as a USB drive) for use in the laboratory classes. Alternatively you may choose to email a copy of your work to yourself if you do not have such a device. Students are responsible for their own storage of work. Ensure regular backups are made, as no exceptions will be given for loss of files

Principal material covered in this subject will be available from the subject's web site. Students are free to print their own copies of these slides. These slides may then be combined with notes written during lectures to form the principal introductory programming reference for this subject. For course revision, students should review each lecture and complete any homework tasks.

In addition to scheduled class sessions, students are expected to spend additional time in individual study and research. As a general guideline students will need to spend at least 1 hour in private study (including completing homework and revision) for every two hours of scheduled class time.

Teachers will be available for a consultation time each week. Students will be notified of the time and location of the consultation session during Week 1 of the Session. It is recommended that students experiencing difficulty with this subject arrange to consult with the teacher as difficulties are encountered.

Some students may require tutorial support to improve language/literacy skills. Where this is recommended, students will be advised to use the College's self-access facilities in the Multimedia Centre in their own time. These facilities are located upstairs in building 30.

Learning resources

The following reference texts are optional:

D. S. Malik, *C++ Programming: Program Design Including Data Structures*, Thompson Learning, 2002

John R. Hubbard, *Programming with C++*, Second Edition, Schaum's Outlines Series, McGraw-Hill, 2000

Walter Savitch, *Problem Solving with C++: The Object of Programming*, Third Edition, Addison-Wesley, 2001

Bjarne Stroustrup, *The C++ Programming Language*, Third Edition (Hard Cover), Addison-Wesley, 2000

Forouzan and Gilberg, *Computer Science: A Structured Programming Approach Using C++*, Nelson Learning, 2000

Subject outcomes

Successful completion of Procedural Programming will enable students to:

- ▶ Analyse and explain the behaviour of simple programs.
- ▶ Apply basic C++ functionality to code simple algorithms.
- ▶ Display a working knowledge of good programming style.
- ▶ Design, implement, test and debug simple programs.
- ▶ Apply features of an integrated development environment to develop programs.
- ▶ Apply the techniques of structured decomposition to break a program into smaller pieces.
- ▶ Further develop C++ practical skills by writing computer programs.
- ▶ Demonstrate effective communication skills with other C++ programmers.
- ▶ Undertake programming and computing-related subjects at University.

Subject outline in weeks

The following guide to lessons and activities may be adjusted to suit the needs of the group as long as subject outcomes and assessment criteria are met.

Generally, the subject will be divided into four modules with each module comprised of lectures, an assignment and a test.

Week 1 - Background of Computing and Programming

History of computers, Algorithms, Programming style.

Week 2 - Introduction to C++ Programming

C++ Data types, The Representation of Data, Binary numbers, Operators, Simple C++ programs and Stream I-O.

Week 3 - C++ Selection Control Structures

The *if* and *switch* statements.

Ass1 Due - 5 marks

Week 4 - C++ Loop Control Structures

The *for* loop, *while* loop and *do-while*, loop. *break* & *continue* statements.

Test 1 - 7.5 marks

Week 5 - C++ Libraries and Arrays

C++ Libraries, Array data types

Week 6 - Strings and Formatted I-O

Character Arrays, Strings, I-O streams, Format manipulators

Ass 2 Due - 5 marks

Week 7 - Functional Decomposition I

Intro to Functions and Functional decomposition

Test 2 - 7.5 marks

Week 8 - Functional Decomposition II

Advanced Functions - pass by reference, pass by value, passing arrays to functions.

Week 9 – Storage Structures and User Defined Data Types

Structs, Unions, Enums and User Defined Data Types

Ass 3 Due - 5 marks

Week 10 - File I-O, Debugging Strategies

C++ File streams, Formatted I-O with files, Debugging Strategies.

Test 3 - 7.5 marks

Week 11 - Multi-File Programs, Recursion and Recursive Algorithms

Multi-file programs, Projects and Header Files, Recursion and recursive algorithms

Ass 4 Due - 5 marks

Week 12 - Revision

Revision

Test 4 - 7.5 marks

Weeks 13 & 14 Final examination Period

Examination and study period. Please refer to examination timetable for the exact date, time and location of the final exam.

Assessment

Assessment and plagiarism policy

All written assessment tasks, with the exception of examinations and in-class tasks, must be word-processed unless students are otherwise advised.

Students must keep copies of all assessment tasks submitted for marking with the exception of class tests and examinations.

Plagiarism is a form of cheating or stealing that happens when a student uses someone else's work and presents it as his/her own without showing where it comes from. To avoid this, students are expected to submit their own original work for assessment and to accurately acknowledge all references and sources used in essays and assignments.

For information regarding assessment, plagiarism, acknowledging sources and examination rules, please refer to the Wollongong College Australia Student Handbook

<http://www.wca.uow.edu.au/handbook>

Assessment Schedule

Task	Due	Weighting	Length/Time
Assignments	Weeks 3, 6, 9, 11	20% (4 x 5)	Various
Module Tests	Weeks 4, 7, 10, 12	30% (4 x 7.5)	90 minutes
Final Examination	Week 13/14	50%	3 hours

Note: A final mark of 50% or higher is required to pass ALL Diploma subjects.
A mark between 45% and 49% is NOT a pass.

Marking Guidelines

WCA best practice is that students can normally expect to have assessment tasks handed back within two weeks, and before the next assessment task is due. On occasion there may be exceptions to this time frame due to, for example, the size of the task, the size of the class, teacher illness or teacher leave.

Where there are several teachers marking a major assessment task, tasks will be handed back by all the teachers within the same week.

Assessment criteria and explanation of components

Subject delivery and assessment is based on 4 modules worth 12.5 marks each and a final examination worth 50 marks giving a total out of 100 marks. Each module is comprised of an Assignment (5 marks) and a Test (7.5 marks).

Assignments 1,2,3 and 4

4 X 5%

Students will design and code C++ programs according to provided program specifications.

Assignment 1 (Week 3) – Will be assessing topics from module 1 (weeks 1 – 3).

Assignment 2 (Week 6) – Will be assessing topics from module 2 (weeks 4 – 6).

Assignment 3 (Week 9) – Will be assessing topics from module 3 (weeks 7 – 9).

Assignment 4 (Week 11) – Will be assessing topics from module 4 (weeks 10 – 11).

Please note that each module builds on the knowledge in the previous module(s), hence to complete assignments 2 – 4 concepts from the preceding modules are assumed knowledge.

Assignments will require work to be completed on the computer. For most students the practical classes (3 hours per week) will provide sufficient time to complete these tasks. If additional computer time is required it will be the responsibility of students to make use of University computer laboratories in their own time. The laboratory schedule and availability can be viewed in the laboratory.

All assessable tasks must be the student's own work. Group work on any assessable task is not permitted. Copying another person's code, or allowing someone else to copy your work, is regarded as cheating. Any student involved in cheating will receive no marks for the work. If a student is having difficulty understanding or completing any task, the student should not hesitate to ask for assistance. (This is, of course, AFTER THE STUDENT HAS EARNESTLY TRIED TO COMPLETE THE TASK).

Assignments must be submitted electronically (in full); this procedure will be explained to you by your lecturer. Please ensure you label the assignments with the following information typed clearly on the label and on the top of each file: student name, student number, subject name, assignment number, the due date. Submitted C++ code files should also have a brief statement describing what is contained in the file. The title block should be appropriately commented so as to not effect compilation of the file. To receive full marks assignments must be demonstrated during the lab class in the following week. Assignments that are not demonstrated will receive a 1 mark deduction. Late assignments should be submitted via e-learning.

Late assignments should be submitted via email to your teacher with the course number, and assignment number clearly entered into the email subject (e.g. **WUCT114 - Ass 2**). Any emailed assignments are sent at the student's own risk. (Return receipts can be enabled on most email applications to receive confirmation.)

Late submission will be penalised in accordance with the student handbook.

1 day late:	10% of mark
2 days late:	20% of mark
3 days late:	30% of mark

(Work submitted later than three days will not be marked)

Extension of the due date (without penalty) will only be granted in exceptional circumstances e.g. death in family, medical condition, etc. Appropriate documentary evidence must be provided. A student requiring an extension in these circumstances should submit a Special Consideration Request Form to the Program Manager, as outlined in the Wollongong College Australia student handbook.

Module Tests 1,2,3 and 4

4 X 7.5%

The module tests will consist of multiple-choice and short answer questions. They are closed-book exam conducted in lab class (*i.e.* no reference material or C++ compilers are permitted).

Module Test 1 (Week 4) – Will be assessing topics from module 1 (weeks 1 – 3).

Module Test 2 (Week 7) – Will be assessing topics from module 2 (weeks 4 – 6).

Module Test 3 (Week 10) – Will be assessing topics from module 3 (weeks 7 – 9).

Module Test 4 (Week 12) – Will be assessing topics from module 4 (weeks 10 – 11).

Please note that each module builds on the knowledge in the previous module(s), hence to complete module tests 2 – 4 concepts from the preceding modules are assumed knowledge.

Final Examination

50%

Students who receive a grade of less than 40% in the final exam may be awarded a fail in this subject regardless of the total final mark.

Non-English speaking background students in the Diploma Programs may use foreign language dictionaries for their final exams. Diploma students who wish to use a dictionary must complete the Dictionary Use Application Form available at reception. This form and the dictionary must be submitted to reception **no later than 5pm Friday week 11** for approval.

Please note the following regulations regarding dictionary use:

- The only dictionaries permitted are language dictionaries, with word to word translations only.
- English-English dictionaries, Electronic dictionaries, Terminology dictionaries, or other are not permitted. The dictionary **must not** include English translations or explanations. Any dictionary that includes English explanations or phrases is not acceptable and will not be approved.