



# 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

### WUCT124

## Object Programming

© Wollongong College Australia. All rights reserved 2009.  
WCA-WUCT124-S0/3

# Object Programming

---

## Subject Description

This subject develops a thorough understanding of program design using data structures. It extends WUCT114 and presents pointers, dynamic memory management and exception handling. Other topics include implementation of Sorting and Searching algorithms including the use of `typedefs` to generalise algorithms; Implementation of data structures: queues, stacks, linked lists, dequeues, trees, heaps; Hashing; random access files; Testing of programs: black and white box testing, and the use of debuggers; Use of multi-file organisation; These concepts will be treated through formal lectures, tutorials, assignments and laboratory sessions employing an object oriented language.

## Subject Structure

Object Programming is delivered in a face-to-face format of 7 (seven) hours per week. The subject consists of 2 x 2 hour lectures, and 3 hours of practical laboratory work 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

The majority of the learning materials used in lectures will be made available from the Object Programming website. Students are welcome to print copies of the materials provided on this site for their personal use. The learning materials provided to students represent a summary of the topics being discussed; lectures cover each topic in significantly more detail. It is important that students attend all lectures and actively participate in class discussions.

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

Malik D. S., *C++ Programming: Program Design including Data Structures*, (3<sup>rd</sup> ed.)  
Thompson Learning, 2007

Hubbard J. R., *Programming with C++*, (2<sup>nd</sup> ed.),  
Schaum's Outlines Series, McGraw-Hill, 2000

Hortstmann, C, Budd, T, *Big C++*,  
Wiley Publishing, 2005

Savitch W., *Problem Solving with C++: The Object of Programming*, (5<sup>th</sup> ed.), Addison-Wesley, 2005

Stroustrup B., *The C++ Programming Language*, (3<sup>rd</sup> ed.),  
Addison-Wesley, 2000

Stallings W., *Operating Systems – Internals and Design Principles*, (5<sup>th</sup> ed.),  
Prentice-Hall, 2005

Deitel, H., & Deitel P., *C++, How to Program*, (5<sup>th</sup> ed.),  
Prentice-Hall, 2005

## Subject Outcomes

Successful completion of Object Programming will enable students to:

1. Use memory management and exception handling in software implementation
2. Use dynamic memory allocation to create and maintain dynamic data structures
3. Identify and describe structured data types, their implementation in C++, and use in problem solving
4. Design, implement, test and debug simple programs.
5. Write programs that display a working knowledge of good programming style.

## 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.

### **Week 1 – Module 1**

Introduction / More C++

### **Week 2 – Module 1**

More C++

### **Week 3 – Module 1**

More C++

### **Week 4 – Module 2**

Pointers and Dynamic Memory

### **Week 5 – Module 2**

Pointers and Dynamic Memory

### **Week 6 – Module 2**

Pointers and Dynamic Memory

### **Week 7 – Module 3**

Classes

### **Week 8 – Module 3**

Classes

### **Week 9 – Module 3**

Classes

### **Week 10 – Module 4**

Data Structures

### **Week 11 – Module 4**

Data Structures

### **Week 12 – Module 4**

Data Structures / Revision

### **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.

# Module Content

## Module 1: More C++

- WUCT114 Revision
  - o typedef, type-casting, enumerations
- Multiple Source Programs
  - o header/implementation files
  - o avoiding redefinition of included files
- Preprocessor Directives
  - o definitions, conditional commands, include
- Understanding runtime errors
  - o Segmentation Faults
  - o Bus Errors
- Bitwise operators
  - o and/or/xor/complement/shift operations
- Binary I/O
  - o seekg/seekp
  - o read/write

## Module 2: Pointers and Dynamic Memory

- Pointers
  - o sizeof
  - o address operator (&)
  - o passing function arguments by reference/value
  - o reference operator (\*)
  - o arrays as pointers
  - o pointer arithmetic (p++, etc)
  - o void\* pointers
  - o pointers to structs/classes (-> operator)
  - o function pointers
  - o type-casting
- Dynamic Memory
  - o new operator
  - o memory leaks
  - o delete operator
  - o delete[] operator

## Module 3: Classes

- Classes
  - o classes vs. objects
  - o private/public
  - o accessor/mutator functions
  - o constructors
  - o copy constructors
  - o destructors
  - o static const member variables
- Namespaces
- Other classes
  - o string class
- Exceptions
  - o try
  - o catch
  - o badalloc
  - o catch(...)
  - o Black/White-box testing

## Module 4: Data Structures

- Sorting/searching
  - o Selection/Bubble/Insertion
  - o Binary search
  - o Quicksort
  - o Testing efficiency by counting comparisons/swaps
- Generic coding using typedef statements
- Linked lists
  - o C++ implementation of insert/delete, etc.
- Binary Search Trees
  - o C++ implementation of location, traversals, etc.
  - o Storing trees in arrays
- Overview of other collections
  - o doubly-linked lists, stacks, queues, dequeues, priority queues, heaps
- Hashing
  - o Hash functions
  - o Collisions
  - o Occupancy
  - o Insertion/retrieval of data

# 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
Assignments – 4 in total	Every 2-3 weeks throughout session	25%
Class Tests – 4 in total	Every 2-3 weeks throughout session	20%
Tutorial Work	Continuous	5%
Final Examination	Undertaken in exam. period, weeks 13-14	50%

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

All assessment components are marked according to set marking criteria.

### Assignments

25%

Each course module will include a practical assignment involving the creation of a C++ program which utilises the language features learned within that module.

Assignments are to be submitted on or before the due time. Assignments which are submitted after the due time will be marked as late and will be penalised as outlined in the student handbook. Assignment submissions more than 3 days late will be awarded a mark of zero. Each of the 4 assignments is worth 6.25% of your final mark for this subject.

### Class Tests

20%

Each course module concludes with a written test assessing the student's understanding of the module content in a manner which prepares them for the final examination. Each of the 4 class tests is worth 5% of your final mark for this subject.

### Final Examination

50%

The final examination will be held during the formal examination period in weeks 13 and 14, and will examine all topics covered during the session. 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.

### Class Tests

5%

Throughout the session, small coding exercises will be given to help students consolidate their understanding and put the theory learned into practise.