

# COMP 2017 PROGRAMMING TECHNIQUES (WSTC)

**Credit Points** 10

**Legacy Code** 700257

**Coordinator** Kritika Gupta ([https://directory.westernsydney.edu.au/search/name/Kritika Gupta/](https://directory.westernsydney.edu.au/search/name/Kritika%20Gupta/))

**Description** This subject is intended as a second subject of study in programming. It builds on a basic understanding of procedural programming as would be developed in a first subject. This subject continues the development of programming skills and methodologies required for professional programming and for further study in later computing subjects. Topics covered include multi-dimensional arrays, file I/O, searching and sorting, and an introduction to object-oriented programming involving classes and inheritance

**School** Computer, Data & Math Sciences

**Discipline** Programming

**Student Contribution Band** HECS Band 2 10cp

Check your fees via the Fees ([https://www.westernsydney.edu.au/currentstudents/current\\_students/fees/](https://www.westernsydney.edu.au/currentstudents/current_students/fees/)) page.

**Level** Undergraduate Level 2 subject

**Pre-requisite(s)** COMP 1006

**Equivalent Subjects** COMP 2015 - Programming Techniques LGYA 5800 - Programming Principles 2

**Incompatible Subjects** COMP 2016 - Programming Techniques (Advanced)

## Restrictions

Students must be enrolled at Western Sydney University, The College. Students enrolled in the Extended Diploma programs must have passed 40 credit points in order to enrol in this subject. Students enrolled in the combined Diploma/Bachelor programs listed below must pass all College Preparatory subjects listed in the program structure before progressing to the Year 2 subjects.

## Learning Outcomes

On successful completion of this subject, students should be able to:

1. With a chosen programming language in mind, analyse a given problem and: (a) Develop an algorithm that applies structured programming techniques such as sequence, selection, iteration and modularisation that solve the given problem; (b) Choose suitable data types to store relevant data for the given problem; (c) Implement the solution algorithm using the chosen programming language, data types and control structures; (d) Test and debug the program code to produce a working computer program.
2. Write and implement programs that use data structures such as arrays to solve problems in programming involving multiple data items.
3. Demonstrate how different searching and sorting methods operate and be able to implement them in working computer programs.
4. Store, retrieve and manipulate data programmatically from secondary storage.

5. Use object-oriented methodology to analyse relatively simple problems and develop object-oriented computer program solutions.

## Subject Content

1. Revision and extension of procedural programming structures including:
  - (a) Data types
  - (b) I/O statements
  - (c) Calculations
  - (d) Problem solving techniques
  - (e) Decision making constructs
  - (f) Repetition structures
  - (g) Writing and using functions
  - (h) One-dimensional arrays
2. Multi-dimensional arrays
3. Character strings and textual data
4. Sorting and searching algorithms: sequential search, binary search, selection sort, bubble sort
5. Object-Oriented Programming including:
  - (a) Introduction to classes and objects
  - (b) Class construction
  - (c) Constructors and destructors
  - (d) Inheritance
  - (e) Polymorphism
6. Data files

## Assessment

The following table summarises the standard assessment tasks for this subject. Please note this is a guide only. Assessment tasks are regularly updated, where there is a difference your Learning Guide takes precedence.

Type	Length	Percent	Threshold	Individual/ Group Task	Mandatory
Practical Exercises	2 – 4 week duration	30	N	Individual	
Two practical deliverables – 1 to 5 programming tasks per practical deliverable. Mostly developed in-class over 2 to 4 week period (15% per deliverable)					
Programming Project One deliverable. Typically the solution files will be approximately 500 to 700 lines of code.	500 – 700 lines of code	40	N	Individual	

Practical 90 mins 30 Y Individual  
Programming  
Test  
(Partial  
open book)

Prescribed Texts

- Gaddis. T. (2013). Starting out with Java : from control structures through objects (5th Ed.). Boston : Pearson