# CS325-15 Compiler Design

## 20/21

**Department**
Computer Science
**Level**
Undergraduate Level 3
**Module leader**
Gihan Mudalige
**Credit value**
15
**Module duration**
10 weeks
**Assessment**
Multiple
**Study location**
University of Warwick main campus, Coventry

---

## Description

### Introductory description

A compiler is a program that can read a program in one language - the source language - and translate it into an equivalent program in another language - the target language

### Module aims

The module will provide a through introduction to the principles of compiler design, with an emphasis on general solutions to common problems as well as techniques for putting the extensive theory into practice.

### Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

- Languages and Grammars: regular expressions, context-free grammars, BNF.
- Parsing: top-down and bottom-up techniques.
- Semantic Analysis: attribute grammars, translation schemes, type inference, symbol tables. Code Generation: run-time environment, intermediate code, register allocation, optimization. Programming Paradigms: issues in the compilation of imperative, functional, and object-oriented languages

### Learning outcomes

By the end of the module, students should be able to:

- A successful student will have acquired the skills to understand, develop, and analyse recognizers for programming languages. The student will also be able to deploy efficient and methodical techniques for integrating semantic analysis into the afore-mentioned recognizers, and generate low-level code for most constructs that characterise imperative and functional programming languages.

### Indicative reading list

(a) Appell, Modern Compiler Implementation in Java, Cambridge University Press, 2003
(b) Watt and Brown, Programming Language Processors in Java, Prentice Hall, 2000
(c) Grune, Bal, Jacobs, and Langendoen, Modern Compiler Design, Wiley, 2000.
(d) Aho, Sethi and Ullman, Compilers Principles, Techniques and Tools, Addison-Wesley.

### Subject specific skills

Develop an end-to-end compiler. Use of modern and industrial-grade compiler development software, techniques and tools.

### Transferable skills

Technical - Programming. Following online tutorials. Version control and software development. Software testing and debugging.
Creativity - Designing tangible and strategic solutions (compilers).
Multitasking - Time management, organisation skills and meeting deadlines.
Critical thinking - Problem-solving, analysis of possible solutions.
Communication - Listening, writing, technical communication skills

---

## Study

## Study time

| Type | Required |
|---|---|
| Lectures | 30 sessions of 1 hour (20%) |
| Seminars | 5 sessions of 1 hour (3%) |
| Practical classes | 5 sessions of 1 hour (3%) |
| Private study | 110 hours (73%) |
| Total | 150 hours |

### Private study description

Revision of lecture notes
Private study of online tutorials and coursework
Background reading - reading recommended textbook sections
Attempting/doing past exam paper questions

## Costs

No further costs have been identified for this module.

---

## Assessment

You do not need to pass all assessment components to pass the module.

Students can register for this module without taking any assessment.

### Assessment group D1

| | Weighting | Study time |
|---|---|---|
| Unsupervised practical assignments | 40% | |
| 2 hour examination (Summer) | 60% | |
| Exam | | |

~Platforms - AEP

---

- Online examination: No Answerbook required

### Assessment group R

| | Weighting | Study time |
|---|---|---|
| CS325 Examination | 100% | |
| CS325 resit examination. | | |

~Platforms - AEP

### Feedback on assessment

Written feedback on coursework.

Past exam papers for CS325

---

## Availability

### Pre-requisites

Students must have studied the material in CS126 or CS259 or equivalent subject material.

## Courses

This module is Optional for:

- Year 3 of UCSA-G4G1 Undergraduate Discrete Mathematics
- Year 3 of UCSA-G4G3 Undergraduate Discrete Mathematics
- Year 4 of UCSA-G4G2 Undergraduate Discrete Mathematics with Intercalated Year

This module is Option list A for:

- Year 3 of UCSA-G400 BSc Computing Systems
- Year 4 of UCSA-G504 MEng Computer Science (with intercalated year)
- Year 3 of UCSA-G500 Undergraduate Computer Science
- Year 4 of UCSA-G502 Undergraduate Computer Science (with Intercalated Year)
- Year 3 of UCSA-G503 Undergraduate Computer Science MEng

This module is Option list B for:

- Year 4 of UCSA-G401 BSc Computing Systems (Intercalated Year)
- Year 3 of UCSA-G402 MEng Computing Systems
- Year 4 of UCSA-G403 MEng Computing Systems (Intercalated Year)
- Year 3 of UCSA-G406 Undergraduate Computer Systems Engineering
- Year 3 of UCSA-G408 Undergraduate Computer Systems Engineering
- Year 4 of UCSA-G407 Undergraduate Computer Systems Engineering (with Intercalated Year)
- Year 4 of UCSA-G409 Undergraduate Computer Systems Engineering (with Intercalated Year)
- Year 3 of USTA-G302 Undergraduate Data Science
- Year 3 of USTA-G304 Undergraduate Data Science (MSci)
- Year 4 of USTA-G303 Undergraduate Data Science (with Intercalated Year)
- UMAA-G105 Undergraduate Master of Mathematics (with Intercalated Year)
    - Year 3 of G105 Mathematics (MMath) with Intercalated Year
    - Year 5 of G105 Mathematics (MMath) with Intercalated Year
- Year 3 of UMAA-G100 Undergraduate Mathematics (BSc)
- UMAA-G103 Undergraduate Mathematics (MMath)
    - Year 3 of G103 Mathematics (MMath)
    - Year 4 of G103 Mathematics (MMath)
- UMAA-G106 Undergraduate Mathematics (MMath) with Study in Europe
    - Year 3 of G106 Mathematics (MMath) with Study in Europe
    - Year 4 of G106 Mathematics (MMath) with Study in Europe
- Year 4 of UMAA-G101 Undergraduate Mathematics with Intercalated Year