



Course Outline (Higher Education)

School:	School of Engineering, Information Technology and Physical Sciences
Course Title:	COMPARATIVE PROGRAMMING LANGUAGES
Course ID:	ITECH5403
Credit Points:	15.00
Prerequisite(s):	(ITECH1000 or ITECH1400)
Co-requisite(s):	Nil
Exclusion(s):	Nil
ASCED:	029999

Description of the Course:

The course content consists of a comparative study of several high-level programming languages such as C, C++, Java, Python, Perl, Lisp, Prolog, Ruby, and Smalltalk. The languages are compared with regard to their intrinsic data types and operations, control structures, implementation, and ease of use for various programming problems, allowing developers to choose the best programming language to use for a variety of different tasks.

Grade Scheme: Graded (HD, D, C, P, MF, F, XF)

Work Experience:

No work experience: Student is not undertaking work experience in industry.

Placement Component: No

Supplementary Assessment: Yes

Where supplementary assessment is available a student must have failed overall in the course but gained a final mark of 45 per cent or above and submitted all major assessment tasks.

Program Level:

Level of course in Program	AQF Level of Program					
	5	6	7	8	9	10
Introductory	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Intermediate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓	<input type="checkbox"/>
Advanced	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Learning Outcomes:

On completion of the course, students should be able to: *Easily learn a new programming language by systematically investigating its definition and implementation, *Identify and use the most appropriate language or language feature to solve a particular problem, *Appreciate the relationship between language definition and its execution time characteristics, *Ability to select a domain specific language to solve a particular problem.

Knowledge:

- K1.** Explain advanced programming concepts,
- K2.** Describe historical and current issues in the design, implementation and application of programming languages,
- K3.** Analyse the influence of comparative language design and implementation on programming practice and methodology.
- K4.** Compare and contrast features across various programming languages and their implementation strategies,

Skills:

- S1.** Demonstrate ability to make a comparative assessment of language design and implementation,
- S2.** Analyse suitability of various languages for differing tasks and environments,
- S3.** Appraise various programming paradigms, including those of representative languages
- S4.** Appraise and implement strategies appropriate for different programming languages
- S5.** Demonstrate ability to write programs using a number of different modern programming languages.

Application of knowledge and skills:

- A1.** Determine the most suitable programming language for a variety of different tasks, based on the requirements and characteristics of the particular task being undertaken.

Course Content:

The course consists of a comparative study of several high-level programming languages such as C, C++, Java, Python, Perl, Lisp, Prolog, Ruby, and Smalltalk. The languages are compared with regard to their intrinsic data types and operations, control structures, implementation, and ease of use for various programming problems.

Topics may include:

- Evolution of major programming languages
- Comparative understanding of procedural, object-oriented, functional and logical programming languages,
- Lexical and Syntax Analysis
- Names, Bindings and Scopes,
- Data types
- Abstract data types, encapsulation and object-orientation
- Expressions and assignment statements
- Statement level control structures
- Dynamic programming, introspection, reflection and meta-programming
- Fundamentals and design considerations of subprograms and concurrency
- Exception and event handling

Values:

- V1.** Learn to think more broadly about programming and programming languages
- V2.** Enable students to understand current trends in programming language research

- V3.** Appreciate the history and diversity of ideas in programming languages
- V4.** Develop a strong appreciation for current trends in programming languages and research

Graduate Attributes

The Federation University FedUni graduate attributes (GA) are entrenched in the [Higher Education Graduate Attributes Policy](#) (LT1228). FedUni graduates develop these graduate attributes through their engagement in explicit learning and teaching and assessment tasks that are embedded in all FedUni programs. Graduate attribute attainment typically follows an incremental development process mapped through program progression. **One or more graduate attributes must be evident in the specified learning outcomes and assessment for each FedUni course, and all attributes must be directly assessed in each program**

Graduate attribute and descriptor		Development and acquisition of GAs in the course	
		Learning Outcomes (KSA)	Assessment task (AT#)
GA 1 Thinkers	Our graduates are curious, reflective and critical. Able to analyse the world in a way that generates valued insights, they are change makers seeking and creating new solutions.	K1 - K4, S1, S2, S5, A1	1 and 2
GA 2 Innovators	Our graduates have ideas and are able to realise their dreams. They think and act creatively to achieve and inspire positive change.	K3, S1, S2, S5, A1	1 and 2
GA 3 Citizens	Our graduates engage in socially and culturally appropriate ways to advance individual, community and global well-being. They are socially and environmentally aware, acting ethically, equitably and compassionately.	Not applicable	Not applicable
GA 4 Communicators	Our graduates create, exchange, impart and convey information, ideas, and concepts effectively. They are respectful, inclusive and empathetic towards their audience, and express thoughts, feelings and information in ways that help others to understand.	K1, K2, S1, S3, S4	1 and 2
GA 5 Leaders	Our graduates display and promote positive behaviours, and aspire to make a difference. They act with integrity, are receptive to alternatives and foster sustainable and resilient practices.	Not applicable	Not applicable

Learning Task and Assessment:

Learning Outcomes Assessed	Learning Tasks	Assessment Type	Weighting
K1, K2, K4, S1, S2, S3, S4, S5, A1	Students will demonstrate the ability to implement programs in a range of programming languages and illustrate the analytical skills to compare and contrast language features for different tasks and environments.	Assignments	65 - 75%
K1, K2, K3, S2, A1	Students will demonstrate the ability to explain the history, issues and theory of programming concepts and language design with the purpose of determining the most suitable language for various tasks and environments.	Examination/Test(s)	25 - 35%

Adopted Reference Style:

APA

Refer to the [library website](#) for more information

Fed Cite - [referencing tool](#)