

CSCU9Y4 Programming Language Paradigms

Lecturers

Prof C Shankland (ces@cs.stir.ac.uk) (Course Organiser)

Dr M Fayed (mmf@cs.stir.ac.uk)

Prerequisites

CSC9A3 / CSC933 / CSC993 / CSCU9A3

Credits

20 credits at SCQF level 9

Learning Outcomes

The student should know and understand:

- The principal programming language concepts and the way they are dealt with in the imperative (procedural and object-oriented) and declarative (functional and logic) language paradigms
- The underlying principles behind language design and that superficially very different languages are often, in fact, very similar
- The link between those underlying principles and their implementation, including an understanding of the runtime system of common languages and memory allocation techniques
- How modules and classes can be used to structure large programs and how they provide abstraction, information hiding and encapsulation
- The central role of data structures in language design
- The use of regular expressions to capture patterns
- How to employ simple formal grammars for sentence generation and parsing
- Application of theory and techniques to unseen problems without reference to notes, to work independently and under a time constraint.

Transferable Skills

- Ability to plan work, to understand how tasks can be specified, to undertake independent creative activity and to bring it to a successful conclusion.
- Ability to organise and present technical material in written form.

Contents

- The imperative (procedural and object-oriented) language paradigm including: types, variables, declarations, expressions, statements, pointers, procedures, methods, parameter passing, modules and classes.
- Inheritance and dynamic binding in the object-oriented language paradigm.
- Storage allocation: run-time stack, heap storage and garbage collection.
- Data abstraction and libraries, the hiding of implementation detail in collections.
- Generic type constructs and associated language features.
- Definition of syntax and semantics.
- The declarative (functional and logic) language paradigms including: type inference, polymorphism.
- Scripting languages, processing strings and the impact of the Internet on language design.
- Reinforcement of language concepts through practical exercises (in Java, C, Prolog, and a functional language)

Assessment

Lab checkpoints (20%), a writing and peer assessment assignment (30%), and an examination (50%).

Textbooks

See the online reading list.

Requirements

To be eligible for a pass grade in the module, students must:

- Complete at least one lab checkpoint
- Submit the assignment
- Attempt the examination.

Students who do not fulfil these requirements will be awarded grade X for the module.

[See paragraph 61 of <http://www.stir.ac.uk/regulations/undergrad/assessmentandawardofcredit/>]

Coursework extensions

Students who can show good cause may be permitted extensions to coursework deadlines. "Good cause" may include illness, for which a medical certificate or other evidence will be required. Students must request an extension by contacting the module coordinator, supplying relevant evidence, no later than seven days after the published assignment deadline. Extensions will only be granted for acceptable reasons, and will not normally be beyond such time as solutions and feedback are returned to the rest of the class.

[See Section 6.2.6 of <http://www.stir.ac.uk/academicpolicy/handbook/assessment/>]

Late submission of coursework

Coursework that is submitted late will be accepted up to seven days after the published deadline (or expiry of any agreed extension) but the mark will be lowered by three marks per day or part thereof. After seven days the piece of work will be deemed a non-submission. [See paragraphs 66-68 of

<http://www.stir.ac.uk/regulations/undergrad/assessmentandawardofcredit/>]

Repeat assessments

Students who fulfil the module requirements, but obtain an overall fail mark (0-39) following the Main examination, are eligible for repeat assessment. In most cases, this will be a repeat examination (grade RE). In some cases, the Examiners may permit repeat submission of coursework (grade RC) or a repeat of both coursework and examination (grade RA). Repeats are not permitted for laboratory checkpoints or group work. The mark following any repeat assessment is capped at 40.

Discretionary repeat assessments

In exceptional circumstances, a student who has not met all the module requirements, following the Main examination, may be permitted a discretionary repeat. This may be a repeat examination (grade XE), repeat submission of coursework (grade XC), or a repeat of both coursework and examination (grade XA). Repeats are not permitted for laboratory checkpoints or group work. The mark following any repeat assessment is capped at 40. If you are granted a discretionary repeat assessment but do not attempt it, you will be awarded grade X for the module.

In deciding whether to grant a discretionary repeat, the Examiners will consider your record of attendance and engagement in the module. Students with a poor attendance record will not normally be permitted a discretionary repeat.

Deferred examinations

If you are unable to attend the Main examination, you must apply for a Deferred examination through the link on the Student Portal. If a Deferred examination is denied and you do not attend the Main examination, you will be deemed not to have met the module requirements.

[See <http://www.stir.ac.uk/register/studentinformation/exams/deferredexams/>]

Attendance recording

Attendance at lectures, tutorials and practical classes will be recorded. If you are unable to attend a class, please email the module organizer, and submit a self-certification of absence via the Portal if appropriate. The University has a policy of monitoring attendance. Repeated absence will be followed up in order to identify any problems at an early stage and to offer students appropriate support.

Students should note that failure to attend classes (lectures, tutorials or practicals) severely jeopardises your chances of passing the module.

- **Tutorials** are of paramount importance in this course. You must prepare answers in advance of the tutorial hour, and are expected to participate fully in the discussion in class. This is **essential**.
- Participating in **practicals is necessary** in order to obtain checkpoints.
- Failure to attend classes (lectures, tutorials or practicals) severely jeopardises your chances of passing the module.

CSCU9Y4 Schedule

Lectures: The first lecture will be on Tuesday 17th January 2017 in LTA5 at 1100.

Thereafter, lectures will be:

Day	Time	Room
Tuesday	1100	LTB3 (LTA5 in week one only)
Thursday	1000	LTA1

Note the schedule below. Lectures are not planned in every slot.

Reading material will be specified for each lecture and students are expected to be familiar with this material *before* the lecture.

Tutorials

One tutorial per week.

Tuesday	1500	3B139
Thursday	0900	3B147
Thursday	1200	3B139

Tutorials will commence on **Tuesday 24th January 2017**.

A problem sheet will be made available before the tutorial class and students are expected to attempt *all* problems before coming to the tutorial.

Practicals

One practical per week on Tuesday at 1400 in 1A11.

Additional catch-up practical per week on Thursday at 1600 in 1A13.

The first practical is on **Tuesday 17th January 2017**. See the schedule below – practical classes do not run every week. A work sheet will be made available before the practical class and students are expected to attempt much of this before coming to the practical. The practical hour is for students to access help from tutors and demonstrators.

Assignment Details

Checkpoints: 20%. Written report plus peer assessment: 30%. Multiple deadlines for the latter in the week of 13th March 2017.

Schedule

Note that the Schedule is subject to revision during semester, according to progress with each element.

	PRACTICAL	TUTORIAL	LECTURE	LECTURE
	Tues 1400 (1A11) Thurs 1600 (1A13)	Tues and Thurs	Tues 1100 (LTA5 [1] LTB3 [2-5,7-12])	Thur 1000 (A1)
Week of Jan 16	Prac 1 – Java 1 (mmf)		overview (ces)	execution model (ces)
Week of Jan 23	Prac 2 - Java 2 (mmf)	Tut 1 - design	organisation (ces)	types 1 (ces)
Week of Jan 30	Prac 3 - Java 3 (mmf)	Tut 2 - organisation	types 2 (ces)	-
Week of Feb 6	Prac 4 - Java 4 (mmf)	Tut 3 - bindings	references 1 (ces)	-
Week of Feb 13	Prac 5 - refs 1 (mmf)	Tut 4 - references	parameter passing (ces)	syntax and semantics 1 (ces)
Week of Feb 20	reading week	reading week	reading week	reading week
Week of Feb 27	Prac 6 – refs 2 (mmf)	Tut 5 - writing (to be confirmed)	syntax and semantics 2 (ces)	declarative overview (ces)
Week of Mar 6	Prac 7 - Prolog 1 (mmf)	Tut 6 - syntax and semantics	Prolog (ces)	-
Week of Mar 13	Prac 8 - Prolog 2 (mmf)	Tut 7 - Prolog	(peer assessment session 0900-1100 in 2B85)	functional prog 1(ces)
Week of Mar 20	Prac 9 - FP 1 (mmf)	Tut 8 - functional programming	functional prog 2 (ces)	web and scripting (ces)
Week of Mar 27	Prac 10 - FP 2 (mmf)	Tut 9 - web and scripting	-	-
Week of Apr 3	-	-	revision lecture on exam technique	-

Prof Shankland has to miss the Thursday lectures 2nd and 9th February. They may be rescheduled.

Peer assessment session: Tuesday March 14th at 0900-1100 in 2B85

Depending on progress, free slots (marked “-“) may have classes allocated later in the semester.