

SYLLABUS: PROGRAMMING LANGUAGE CONCEPTS

COURSE & INSTRUCTOR INFORMATION

Course

Course Prefix, Number, Section, Title: COSC 30403-045 Programming Language Concepts

Semester and Year: Spring 20XX

Number of Credits: 3

Course Component Type: Lecture

Class Location: XXX

Class Meeting Day(s) & Time(s): XXX

Instructor

Instructor Name: Dr. XXX

Office Hours: XXX

Preferred Method of Contact: Email

Email:

Response Time: Emails will be responded within 24 hours including weekends.

Final Evaluative Exercise & Important Dates

FINAL EXAM: THURSDAY, MAY 8 11:00 AM – 1:30 PM AT SWR LH4
MIDTERM EXAM (TENTATIVE): THURSDAY, MARCH 6 12:30 – 1:50 PM AT SWR LH4
LAST DAY TO DROP: TUESDAY, APRIL 1
LAST DAY TO SELECT P/NC: MONDAY, APRIL 21

Note for students: The syllabus is your first course reading. It provides an orientation to, overview of the flow, and expectations of the course. You should turn to the syllabus for details on assignments and course policies.

Grader

Hien Dau: hien.minh.dau@tcu.edu for grading students whose last names start with O – Z.

Student Resources & Policy Information

Click or scan QR code for resources to support you as a TCU student. Please note section on [Student Access and Accommodation](#), [Academic Conduct & Course Materials Policies](#), and [Emergency Response & TCU Alert](#).



COURSE DESCRIPTION

Description

A study and comparison of the concepts and constructs of major programming language paradigms. Topics include Programming Language Syntax, Names, Scopes, and Bindings, Semantic Analysis, Control Flow, and Data Types. In addition, the course includes learning ANTLR to write parsers, and studying Scala, a functional programming language.

Prerequisites & Concurrent Enrollment

COSC 20803 and MATH 20123; both with C- or better.

COURSE MATERIALS

Required Material

Textbook:

- Michael L. Scott, *Programming Language Pragmatics*, 4th Edition, ISBN: 978-0124104099
- Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language*, 2nd Edition, ISBN: 978-0131103627
- Martin Odersky, Lex Spoon, Bill Venners and Frank Sommers, *Programming in Scala*, 5th Edition, ISBN: 978-0997148008

Supplementary Resources

Slides: Lecture slides will be made available on TCU Online.

Device Information: Students **MUST** be able to access laptops or desktops to complete all programming projects. The operating system can be Windows or macOS.

LEARNING OUTCOMES

1. Recognize major programming language features.
2. Explain the stages of programming language interpretation and compilation.

3. Explain design concepts, design alternatives and trade-offs, and implementation considerations for scope, binding, data types, expressions, control structures.
4. Understand the principles of language design and implementation.
5. Choose a suitable programming language for a given problem or domain.
6. Evaluate modern programming languages critically.
7. Quickly become productive in a new language.

COURSE REQUIREMENTS

Assessments

Written Assignments:

- There are four written assignments. The questions are selected from the textbook *Programming Language Pragmatics*, 4th Edition.
- Late written assignments are **NOT** allowed and are worth 0 points.
- Students are encouraged to share thoughts and ideas. However, it is **STRICTLY PROHIBITED** to share direct answers.

Programming Projects:

- There are five programming projects. The projects include writing parsers through ANTLR and developing programs in Scala.
- Late programming projects are accepted with a penalty of 10% per calendar day up to a maximum of 5 days. Afterwards, the programming project is worth 0 points.
- Students are encouraged to share thoughts and ideas. However, a student should **NEVER** share his code to another student. It is **PROHIBITED** to even share one line of code.

Presentation:

- Presentations are arranged by groups. Five or six students will form a group and choose a programming language in the middle of the semester.
- Each group will record a presentation to introduce the chosen programming language and submit the recording at the end of the semester.

Exams:

- Both exams are taken in the lecture classroom. Students are asked to handwrite the answers. The final exam is comprehensive with an emphasis on the second half of the semester.
- Both exams are closed book. Each student is allowed to bring one US letter size, double-sided cheat sheet. All information on the cheat sheet must be **HANDWRITTEN**.
- Computers are not allowed. Calculators are not allowed.

- **ONLY** the final exam can be rescheduled. Graduating students **MUST** reschedule the final exam. For non-graduating students, rescheduling is **ONLY** permitted to meet the 24-hour rule.

Grading Philosophy & Policy

Late Work:

- Late written assignments are **NOT** allowed and are worth 0 points.
- Late programming projects are accepted with a penalty of 10% per calendar day up to a maximum of 5 days. Afterwards, the programming project is worth 0 points.

Coupon System:

- Each student has **ONE** coupon for the semester, which can be applied to one of the four written assignments **OR** one of the five programming projects.
- If you receive a low grade for any reason, including but not limited to incorrect submission format, missed submission, incorrect files submitted, or incomplete work, you can submit the updated solution to Coupon Redemption on TCU Online.
- The grader will then regrade your updated solution with a 20% penalty applied.

Make-Up Work:

- Make-up work will **ONLY** be permitted for two reasons: an official university absence or documented medical appointments.
- A student who has an official university absence must contact the instructor **BEFORE THE ABSENCE** to schedule a make-up of any assessment that will be missed.
- Because it is considered an infringement on student privacy for me to have access to student medical records, I cannot accept medical documentation to justify absences. If you have a legitimate reason for your absence and want to provide verification, please access the Absence Documentation Form [here](#). Such medical-related absences will be rescheduled accordingly.

Extra Credits:

- Extra credits/work will **NOT** be given with no exceptions. This policy will be strictly enforced toward the end of the semester.
- The final course grades will **NOT** be curved with no exceptions.

Attendance: Attendance will not be taken. Students will be considered **RESPONSIBLE** for all material presented during their missed lectures.

Cheating and Plagiarism: Students are encouraged to share thoughts and ideas. However, a student should **NEVER** share his code to another student. It is **PROHIBITED** to even share one line of code.

Course Assignments & Final Grade

Assessment	Percent Per Instance	Quantity	Percent
Written Assignments	4%	4	16%
Programming Projects	8%	5	40%
Presentation	8%	1	8%
Exams	18%	2	36%

Grading Scale: The final course grades will **NOT** be curved with no exceptions. Please also be aware that extra work will **NOT** be given with no exceptions.

Grade	Score	Grade	Score
A	≥ 93	C	≥ 73 and < 77
A-	≥ 90 and < 93	C-	≥ 70 and < 73
B+	≥ 87 and < 90	D+	≥ 67 and < 70
B	≥ 83 and < 87	D	≥ 63 and < 67
B-	≥ 80 and < 83	D-	≥ 60 and < 63
C+	≥ 77 and < 80	F	< 60

COURSE SCHEDULE (TENTATIVE)

This schedule represents current course plans. Due to a significant increase in enrollment, the schedule is tentative. Due dates will be communicated and announced during lectures, and the definite due dates will be reflected on TCU Online.

Week	Topic	Due
1	Introduction	
2	Syntax	
3	Syntax	Written Assignment 1
4	Syntax High-Level Languages	Programming Project 1A
5	High-Level Languages	Written Assignment 2
6	High-Level Languages Semantic Analysis	Programming Project 1B
7	Semantic Analysis	Written Assignment 3
8	Semantic Analysis Midterm Exam	
9	Control Flow	

Week	Topic	Due
10	<i>Spring Break</i>	
11	Control Flow Memory Allocation	Programming Project 1C
12	Memory Allocation	Written Assignment 4
13	Data Types Scala Introduction	
14	Functional Objects	Programming Project 2
15	Scala Lists	
16	<i>TBD</i>	Programming Project 3 Group Presentation
17	Final Exam	