I. Course Details

Course Title:

COMP 443: Programming Languages, Spring 2022

Catalog Description:

This course investigates basic concepts of programming languages, including functions, types, and scoping. Functional programming is an emphasis of the course, including first-order functions, lambda expressions, and referential transparency.

Prerequisites:

COMP 222, Introduction to Data Structures and Algorithms

Course Overview:

Purpose:

This course will introduce several programming language concepts in the context of specific languages. This offering will use primarily Scala and Python..

Meeting:

T,Th 2:00 – 3:15 PM, STEM 326

Texts:

Layka and Pollak
Beginning Scala
2nd edition
APress
ISBN: 978-1-4842-0232-6 (ebook) or 978-1-4842-0233-3 (paper)

Optional:
C. Horstmann
Scala for the Impatient
Addison-Wesley
ISBN 978-0-321-77409-5

Web page:

MyGCC

Software:

I will be using IntelliJ in class with Scala, but you may also use the command line and a text editor with Scala and SBT. Python.

Faculty Details:

Instructor:
Dr. Jonathan Hutchins
STEM 340D
hutchinsjo@gcc.edu

Office hours:	M,F 9 – 10:30 am
	T 3:15 – 5:15 pm
	W 12-2:00pm
	Th 9 – 10:00 am

Please don’t hesitate to set an appointment with me outside of office hours. Start a teams chat with me to set a time.

II. ABET Student Outcomes

This course develops students’ ability to do the following:

(SO 6) [CS-specific] An ability to apply computer science theory and software development fundamentals to produce computing-based solutions.

> Students are expected to put into practice what they have learned in previous object-oriented language courses, and to extend that knowledge to new paradigms such as functional programming.

(SO 2) An ability to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.

> Students will craft programs of various scales to familiarize themselves with the concepts from different languages.
>
> Students will gain hands-on experience developing software with different languages and will gain experience learning new languages.

III. Course Outcomes

This course develops students’ ability to do the following:

1. How program elements are constructed from the basic syntactic elements of the languages studied. (supports SO2 and SO6)

2. Major concepts of functional programming. (supports SO6)

3. Differences among the languages studied, including strengths and weaknesses. (supports SO2)

IV. Methodology

Teaching methods:

This course will primarily consist of class lectures, examples, discussions, and hands-on programming activities during class.

Assessment:

Assessment will consist primarily of quizzes/homework, programming projects, a research report and presentation, two semester tests, and a final exam.

Grading:

The available points for the various assignments in this class are:

Progress on in-class exercises	5%
Quizzes/Homeworks	14%
Project average	31%
Language Report/Presentation	9%
In-class exams	25%
Final exam	16%

Final grades will be assigned using the following scale:

A+	97-100
A	93-97
A-	90-93
B+	87-90
B	83-87
B-	80-83
C+	77-80
C	73-77
C-	70-73
D+	67-70
D	63-67
D-	60-63
F	below 60

At the end of the semester, a course average is computed for each student using the percentage weights given above. The scale for assigning letter grades based on the course average is formed by averaging the grading scales of the programs, problem sets, quizzes, and exams using the weights given above. Factors such as attendance, improvement (especially as demonstrated on the final exam), and class participation may be considered when making the final decision in borderline cases.

V. Course Policies

Grading

Grading disputes: From the time an assignment is returned to you with a grade, you have one week to submit a written request for a regrade to the professor. The purpose of a regrade request should be to correct a specific misunderstanding by the grader or to catch something that was marked incorrect but was correct.

Attendance:

Per school policy, three unexcused absences are permitted.

Academic Integrity:

General Policy for COMP Courses:

Students are expected to adhere to the "Honesty in Learning" policy as outlined in the Bulletin and Crimson. Violations will be dealt with as outlined therein. Violations will incur a minimum penalty of 10% of the final course grade or 0% on the assignment, whichever is the larger penalty.

The department's academic integrity policy is designed to encourage each student to learn the material. All work submitted for grading must be the student's own work. For group assignments, all work must be the group's work. Following this policy is not only the expected ethical behavior, it will also prepare you to apply the material on exams and in your career.

You should be able to explain in detail the work that you submit if the professor asks you to do so in person or on an exam.

Example violations of the Honesty in Learning Policy include, but are not limited to, use or possession of material from other people or the Internet that directly and substantially bears on the answer to an assignment; or sharing or publishing copies of solutions that you wrote.

When in doubt, consult the course professor before doing something that may result in a violation of the College's Honesty in Learning policy.
If you need more in-depth assistance on an assignment than is permitted by this standard, meet with the course instructor or attend the weekly tutoring sessions hosted by the ACM.

Specific assignments might have different policies specified in their instructions.

Group Assignments

When an assignment is to be completed in groups, you are free to discuss and share any aspect of that assignment or its solution with the other students in your group. However, the policies above apply to any interaction with students outside of your group (e.g., no discussing solutions with students outside your group).

In-class Electronics Use Policy:

Do not distract other students in class. It is in your best interest to pay attention in class, and you should not hinder others from paying attention.

Course readings:

Depending on your learning style, you may or may not be able to completely understand the material solely from the class lectures. If you find yourself struggling to keep up in class, you should read the textbook sections listed on the schedule **before coming to class**.

Assignments, Exams, and Projects:

All assignments will be distributed via the course's web pages. You are responsible for checking the sites regularly. Due dates for assignments are found on the coursework page.

As stated in the Bulletin, **final exams must be administered according to the time scheduled by the Registrar's office**, and cannot be changed to suit the convenience of the student. It is your responsibility to schedule your travel and work plans accordingly. Students with a GCC excused absence will receive permission to reschedule their final exam. In addition, any student who has three exams scheduled on a given day may request permission from the Dean of the school of the student's first major to reschedule one exam; however, the Dean is not required to grant such a request.

Late work:

Late assignments or projects will be penalized at a rate of 10% of the maximum points possible per day late for the first three days after the due date, late work after three days will not be accepted without instructor approval.

COVID Policy:

Grove City College continues to monitor the COVID-19 outbreak, and we are committed to maintaining a safe and healthy campus community. In order to care for each other, please exercise responsible caution in all academic buildings including in all classrooms, labs, studios, hallways, and restrooms. Additionally, Students showing any symptoms related to COVID-19 (outlined in the Campus Health and Safety Plan) must not attend class. Instead, they should report their condition through the College's COVID-19 portal, work closely with the Office of Student Life and Learning, and not return to class in-person until cleared to do so. During this time, every effort will be made to provide remote learning opportunities. If a student's name does not appear in the COVID-19 portal, faculty are not required to record their lectures or make any other accommodation to make up missed classes or labs. It is recommended that

unvaccinated students wear a mask while attending class. These policies may be modified depending on the pandemic's severity.
If you feel ill, please report your health status in the portal so that the lectures can be provided to you via Teams.

Accessibility & Accommodations:

It is Grove City College's goal that learning experiences be as accessible as possible. If you anticipate or experience physical or academic barriers based on a disability, please let me know immediately so that we may discuss options. You are also welcome to contact the disability services office to begin this conversation or to establish accommodations. The Disability Services Coordinator may be reached at 724-264-4673 or DisabilityServices@gcc.edu.

Counseling Center:

If you are experiencing undue personal or academic stress at any time during the semester or need to talk to someone who can help, you should contact the Counseling Center at 724-458-3788 or email Mrs. Hummel, staff assistant, at mhummel@gcc.edu.

VI. Course Schedule

The schedule outlined below is tentative and may be modified by the instructor throughout the semester to better meet the needs of the class.

Week	Date	Topic	Reading	Due	Quiz
1	1/18	Introduction	p xvii, 1 - 15		
	1/20	Scala basics	p 16 - 27, Ch 2 - while loops		
2	1/25	Scala basics Why Scala?	p 27 - 32, end ch 2 1.2 and 1.3 from "Scala in Action"	Homework (about reading)	
	1/27	Finishing Scala introduction Map[K,V]	p 113 - 114, ch 6: map		
3	2/1	OOP in Scala	p 33 - 53, ch 3	Proj 1a	For-comprehensions, defining a class
	2/3	Pattern matching Traits and mixins	p 73 - 84, ch 5 - pattern matching, p 121 - 132, ch 7		
4	2/8	Functional programming introduction	p 57 - 65, ch4 up to closure	Proj 1b	
	2/10	Functional programming Closures	p 65 - 66, ch 4: closure		
5	2/15	Catch up and review		Homework 2 (program)	Higher-order functions
	2/17	EXAM			

6	2/22	Currying, partially applied functions, and composition Tail recursion	p 67 - 68 p 69 - 70		
	2/24	Options Best practices Comparing Scala, Java, and Ruby PartialFunction	p 219 - 225 p 53-55		
	2/29	BREAK			
	3/2	BREAK			
7	3/7	Scala collections Functional methods of collections	p 91 - 105 p 105 - 109	Proj. 2	
	3/9	Parallel collections Advanced for comprehensions Streams	p 109 - 111 p 111 - 120	Language preferences	
8	3/14	Actors and concurrency			
	3/16	Actors and concurrency Type parameters	p 133 - 143		Functional prog.
9	3/21	Type parameters	p 143 - 145		
	3/23	Catch up and review		Proj 3a	
10	3/28	EXAM			
	3/30	Python introduction	See "Learning Objectives" document for online readings on these topics		
11	4/4	Lists, tuples, classes, and iterators			
	4/6	Sets, dictionaries, and comprehensions		Proj 3b	
12	4/11	Introspection			
	4/13	Parsing			
13	4/18	BREAK			
	4/20	Closures, partials, and function decorators		Homework 3	Python introspection
14	4/25	A taste of Prolog Wrapping up			
	4/27	Language presentations		Language report	
15	5/2	Language presentations		Proj. 4	

Thurs., May 5 Study Day

FINAL EXAM: 6pm, Saturday, May 7

For questions about this syllabus, the assignments, etc., please consult the instructor. Students should not depend on classmates, former class members, etc. to interpret this document, its contents, or other course-related materials. The instructor will have the final word concerning the interpretation of this document and other course-related materials.