

## COMP 348

### Principles of Programming Languages

#### Course Outline – Fall 2020

### Course Information

COMP 348 Principles of Programming Languages

**Instructor and Coordinator:** Ali Jannatpour  
**Method of Contact:** via **Moodle\***  
**Email:** (alij @ ENCS.concordia.ca)

### Schedule:

Lecture (Section U):	Mon-Wed	8:45AM-10:00AM online
Tutorials UA, UB:	Mon	10:15AM-11:05AM online
Tutorials UC, UE:	Wed	10:15AM-11:05AM online
Tutorials UD:	Fri	8:45AM-9:35AM online
Lecture (Section DD):	Wed	5:45PM-08:15PM online
Tutorials DDDA, DDDDB, DDDC:	Tue	2:45PM-3:35PM online
Tutorials DDDD:	Thu	2:45PM-3:35PM online

Office Hours: During class hours via ZOOM

**Method of delivery:** ONLINE; through pre-recorded and live lectures.  
Please check the announcement section on Moodle and follow the instructions on how to attend the live sessions.

Seven PODs will be available remotely via ZOOM. Details will be announced.  
For information on how to attend the online sessions please consult Moodle.

### Course Description

Survey of programming paradigms: Imperative, functional, and logic programming. Issues in the design and implementation of programming languages. Declaration models: binding, visibility, and scope. Type systems, including static and dynamic typing. Parameter passing mechanisms. Hybrid language design. Lectures: three hours per week. Tutorial: one hour per week.

**Pre-requisite:** COMP 249.

## Background

The official pre-requisite for this course is COMP 249. Students are expected to know and remember the knowledge that you acquired in the fundamentals of programming.

The important skills needed to do well in this course include: good programming skills, including the ability to understand, design, implement, and debug programs with non-obvious flow of control.

## Objectives and Learning Outcomes

The objective of this course is to enable students to understand the differences between different programming paradigms, to develop algorithms for problem solving, and finally to develop programs in different programming languages.

The course teaches the principles of programming languages and basic skills needed to understand analyze and develop algorithms for problem solving using different paradigms. Examples from mathematics and data structures will be given and discussed. The following programming paradigms will be covered: logic, functional, procedural, imperative, object-oriented, and aspect-oriented programming.

This course will cover the following topics:

- Survey of programming paradigms: logic functional, Procedural, and Imperative programming.
- Issues in the design and implementation of programming languages.
- Declaration models: binding, visibility, and scope.
- Type systems, including static and dynamic typing.
- Parameter passing mechanisms. Hybrid language design.

## Textbooks

- **COMP 348 Principles of Programming Languages**, 2018 Edition, course pack by Dr. Constantinos Constantinides. This course pack covers the main course material (**required**).
- *Programming Language Pragmatics*, 2009, Michael L. Scott, Elsevier (complementary, optional)

The textbook for the course is: *COMP 348 Principles of Programming Languages*, a course pack by Dr. C. Constantinides, online version available through Concordia Bookstore.

More references and tutorials will be posted, as part of the lecture notes. Lecture notes and other reading materials will be available throughout the semester via Moodle.

## How to Obtain the E-Book

Use the following url to access the e-book, online:

<https://stores.concordia.ca/catalog/product/view/id/58764/s/epack-principles-of-programming-languages-35421e/> (link not available at the moment)

## Teaching Method

The course comprises of weekly lectures and practical training; both in the form of tutorials and assignments. It is emphasized that the attendance in lectures as well as tutorials is mandatory for learning and performing well in this course.

Note that this is an online course. The course is delivered in a semi-synchronous approach which means the content of the course is delivered using pre-recorded videos as well as live sessions. The goal of the live sessions is to deliver a flipped-classroom environment that enables student engagement and active learning. Therefore, you are required to be present during the scheduled hours.

Make sure you are connecting via a device that is equipped with webcam as well as Microphone. Both webcam and microphone are required, especially during demo sessions (see assignments under workload and grading). You also need a reliable highspeed internet connection in order to be able to join the class.

## Method of Contact

Your instructor, as well as all teaching assistants are available on Moodle. Their contact information including zoom meeting IDs are posted on Moodle.

The preferred method of contact is via Moodle messaging system. Please avoid emails. Do not send any attachments by emails. Emails with attachments may be filtered, rejected, or identified as spams.

In addition to Moodle messaging, you are encouraged to use the discussion Forum.

See the important notes section on assignment submission. Do not submit assignments by emails. They will not be marked.

## Course Structure

### Lectures

The lectures are a key component of the course, and you are advised to attend the lectures regularly and attentively. The course material is extensive and includes several difficult concepts. Accordingly, we will try to utilize the lecture time in doing things that you would not get by simply reading the book (explaining difficult concepts, giving you alternate perspectives, relating course material to

other fields, giving you tools to solve problems, etc.) rather than merely repeating facts that you can simply read from the text book. It is strongly advised that you stay current in your reading of the textbook, and attend the lectures regularly. That will enhance your learning experience and prevent you from being lost in the lectures. In fact, we suggest that you casually go through the textbook sections once before the lecture. You are also strongly advised to go and read the material thoroughly after the lecture. Discussing the material with your fellow classmates, solving problems, and asking questions in the lectures are also likely to help you.

This semester, major part of the lectures is delivered as pre-recorded videos. It is important that you watch the pre-recorded videos before attending the live class.

## **Tutorials**

The tutorials will reinforce the material seen during the lectures with examples and exercises. The tutorials are also designed in a way that would assist you to get deeper knowledge and experience of problem analysis and problem solving. It is hence strongly recommended that you attend all the tutorials.

Tutorials will take place every week. During the tutorial session, your tutor will explain more examples as well as discuss the assignments and answer questions related to the course, and more specifically related to the assignments. We strongly encourage you to attend these tutorials.

This semester, Tutorials too are delivered in the form of pre-recorded videos. Therefore, you are required to watch the related videos before attending the tutorial session. The information and links to the videos are posted on Moodle prior to the class. Please consult your schedule to attend your designated tutorial session.

Tutorials start from the second week. There is no tutorial during the first week.

## **PODs (Programmers on Duty)**

In addition to tutorial sessions, PODs are available for additional support. Please consult Moodle for the POD schedule.

There is no POD schedule during the first week.

## **Exams**

We strive hard to give exams where your success does not depend on rote memorization. You will probably find it difficult to answer many of the questions if you have memorized the concepts without understanding them. Moreover, many exam questions will not only test your understanding of concepts, but also your

ability to apply them to solve problems. The assignments should help you in preparing for the exams. There will be no substitution of any missing test.

To pass the course, you must pass the overall exams regardless of your grade in the assignments.

There will be four exams in total. Each exam focuses on one or two programming languages. These exams are timed online activities and there is no replacement or substitution for missed tests under any circumstances. **Please see the important dates at the end of this document.**

Normally they contain some multiple-choice, short coding, and detailed-answer questions. The format of the tests will be discussed prior to the test date.

## **Workload and Grading**

This is a reasonably heavy course with several new programming languages. Therefore, you should be prepared to spend adequate time and effort on this course.

### **Assignments: 20%**

There will be three programming assignments. These assignments are designed to give you more understanding of the course material and to give you practices into various aspects of programming languages. The due dates of the assignments are set and cannot be changed. Check the important dates at the end of this document.

Every assignment covers the programming language / paradigm that is discussed in the class. More information on the specific software, IDE, and compiler version, as well as where it is located will be provided during tutorial times. The details will be given in the class, as well as posted in the assignment document.

Additionally, a couple of reading assignments will be posted throughout the semester to help you further understand the concepts.

Please note that all assignments will be posted on Moodle, all of which must be submitted electronically via Moodle. Not email submission is accepted.

**IMPORTANT:** A demo will be required for each of programming assignments. Markers will communicate with you via Moodle and you must book a demo time with them. You **must** perform the demo (if working in a group, all members of the team must be present during the demo). The demos will be carried out online via ZOOM. More information will be provided by the markers. All members of the team must be present during the demo. Failing to demo the assignment or failing to attend the demo at the reserved time, will result in a 0 mark regardless of your submission. There will be no replacement for a missed demo time.

Note: Due to special circumstances, in case of system glitch or medical reason, a demo may be rescheduled by the marker. In case of rescheduling, up to 40% of the mark may be deducted.

**Exams: 80%**

There will be four exams, each 20%. Tests 1 to 3 will be given during Weeks 4, 8, and 11; whereas Test 4 will be scheduled during the final exam period. See the tentative schedule for more details.

All four tests are pre-scheduled timed exams, outside the lecture hours, and are available for the duration 2 hours.

**Grading:**

The grading of the course will be done based on the relative percentages assigned to the assignments and the tests. For reasons of fairness, we may choose to scale up/down the marks in a particular test or assignment to ensure that all aspects of the course receive a fair weight. Any such "fine-tuning" will be made known to you before the final grades are assessed. Finally, there are no pre-set cutoff points for the final grades; the cutoff points will be decided based on an assessment of difficulty level, class performance, fairness, and instructor's wisdom from teaching and grading the course in the past. That is, there is no definite rule for translation of number grades to letter grades.

- ⇒ In order to pass the course, you must pass the overall grade of all four tests regardless of your grade in the assignments.
- ⇒ There is no standard relationship between percentages and letter grades assigned for this course.

**Graduate Attributes**

As part of both Computer Science and Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of graduate attributes. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) and the Canadian Information Processing Society (CIPS) as being central to the formation of Engineers, Computer Scientists and IT professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. This course aims at teaching and evaluating the following graduate attributes:

## **Demonstrate Knowledge**

Knowledge of imperative and declarative programming paradigms with a wide array of programming technologies such as object-oriented, aspect-oriented, procedural, functional, functional object-oriented, logic programming and multi-paradigm programming. Clauses, procedures and queries, backtracking, the inferencing process, unification, instantiation and resolution. Expressions and lists, binding, defining functions, pure vs. side effecting functions, referential transparency, idempotence, higher-order and anonymous functions and function composition. Primitive vs. composite data types, pointers, arrays, records, unions and enumerated data types. Message passing object-oriented model, inheritance and polymorphism, static vs. dynamic object types, distinguishing between compiler and run-time system responsibilities, casting. Modularity, separation of concerns, crosscutting concerns, join points, point cut designators and advice. Introducing state and behavior into primary units of modularity, context passing, privileged and abstract aspects. Object extensions, modules, and Mixins. Introspection. Generic function object-oriented model, primary vs. auxiliary methods.

## **Problem Analysis**

Analyze problems to determine viable paradigm and language spaces and tradeoffs. Map solutions to specifications. Being able to move from specification to code. Analyze implementations to determine what task they perform and how they do it. Being able to move from code to specification.

## **Use of Engineering Tools**

Deployment of compilers, interpreters, integrated development environments (IDEs) for the implementation and debugging of programs. Being able to extend the functionality of IDEs with plugging. Use of application programming interfaces (APIs) as building blocks, use of libraries to support reuse. Use of frameworks for testing.

The evaluation of these attributes will be based on: 1) Assignments, 2) Tests. This evaluation is used to indicate your proficiency in all of the attributes as per accreditation requirements.

## **Website and other Resources**

Resources for the course (assignments, tutorials, etc.) will be available online on the Moodle; available through the MyConcordia portal <https://moodle.concordia.ca>.

Finally, the faculty web pages have a wealth of information pertaining to our computer systems and software, which includes simple user guides, and answers to many standard questions. You should explore these help pages. Begin your exploration from the URL: <http://www.encs.concordia.ca/helpdesk/faq/faq.php>



## Important Notes

**Contents:** Content belonging to instructors shared in online courses, including, but not limited to, online lectures, course notes, and video recordings of classes remain the intellectual property of the faculty member. It may not be distributed, published or broadcast, in whole or in part, without the express permission of the faculty member. Students are also forbidden to use their own means of recording any elements of an online class or lecture without express permission of the instructor. Any unauthorized sharing of course content may constitute a breach of the Academic Code of Conduct and/or the Code of Rights and Responsibilities

**Backups:** You are advised to retain a copy of all your work until you receive your final grade for the course.

**Software and Hardware Requirements:** All students are expected to have access to a computer with following capabilities:

- Reliable internet connection
- Camera and Microphone (your computer and/or cellphone)
- Document scanning application such as Adobe Scan app (<https://play.google.com/store/apps/details?id=com.adobe.scan.android>)

**Code of Conduct:** Students should be aware of the University's academic integrity and code of conduct as specified in sections 17.10.2 and 17.10.3 in the Undergraduate Calendar: <http://registrar.concordia.ca/calendar/17/17.10.html>

In particular please pay attention to the parts concerning cheating, plagiarism, and possible consequence of violating this code. Sharing codes, design diagrams, algorithms, etc. amongst teams or taken from elsewhere (without proper citation) is not permitted. No need to mention that one learns little from copying others' work. All submissions in this course for assignments must be adequately bound and include as the cover page, the "Expectations of Originality".

**Recommendation:** We highly recommend collaborative learning in this course for better preparation. This is encouraged also when doing the assignments. It is also recommended that you start working on the course assignments as early as possible!

**Submission format:** All assignment-related submissions must be adequately archived in a ZIP file using your ID(s) and last name(s) as file name. The submission itself must also contain your name(s) and student ID(s). Use your "official" name only - no abbreviations or nick names; capitalize the usual "last" name. Inappropriate submissions will be heavily penalized. Only electronic submissions will be accepted. Students will have to submit their assignments (one copy per group) via Moodle. Assignments must be submitted in the right DropBox/folder of the assignments. **Assignments uploaded to an incorrect**



**DropBox/folder will not be marked and result in a zero mark. No resubmissions will be allowed.**

**Using ZOOM:** For both lecture and tutorial sessions you need to use ZOOM software to attend the live session. Registration is not mandatory, however recommended. Use your real name when you connect to the class. Failure to do so, may result in a complete ban. Check course Moodle page for more information on how to connect to the class.

**Exam Platform:** The information on how to take the tests, including the hardware and software requirements, the exam platform, rules, and instructions will be available prior to the exams.

**Oral Exam via ZOOM:** After taking a written test, you may be asked by your instructor to participate in an Oral examination via ZOOM, in which such case your grade for that particular test is subject to passing the oral exam. Failure to do so may result in zero mark.

**Important Notes on Online Exams and Invigilation:** The instructions on how to take the online exams will be available prior to the exam date. It is important that you follow them exactly, before and during the exam. Failure to do so may result in the submitted exam to be invalidated. Please also note the following with respect to online exams and live proctoring software:

- That the tests will take place during the designated date and time. All exam times will be set to Eastern Standard Time. See the important dates at the end of this document.
- That selected tests may be invigilated; and therefore, your image, voice, and screen activity may be recorded throughout the duration of the exam.
- That you must show your Concordia University Identification card to validate your identity. Alternative government-issued photo identification will be accepted, though it is not recommended. Only identification in English or French will be accepted.
- That any recording made (if one is made) will only be viewed by authorized university personnel (no external entity has authorization to review the recording).
- That you will be responsible for ensuring appropriate, properly functioning technology (webcam, a microphone, appropriate browser and an ability to download any necessary software, as well as a reliable internet connection with a minimum of a 3G connection).
- For your online examination(s), you will need to download the appropriate browser lockdown technology and use Zoom. Protocols for entering the examination will be provided by your professor.
- That you should enter the virtual test site and become familiar with the software that will be used for your exam before starting the exam.

- That you will need a quiet place within which to take the exam. Earplugs or noise-cancelling headphones that are not connected to a device may also be used to allow you to focus for the duration of the exam.

**Plagiarism:** The most common offense under the Academic Code of Conduct is plagiarism which the Code defines as “the presentation of the work of another person as one’s own or without proper acknowledgement” (Article 16a). This could be material copied word for word from books, journals, internet sites, professor’s course notes, etc. It could be material that is paraphrased but closely resembles the original source. It could be the work of a fellow student, for example, an answer on a quiz, data for a lab report, a paper or assignment completed by another student. It might be a paper purchased through one of the many available sources. Plagiarism does not refer to words alone - it can also refer to copying images, graphs, tables, and ideas. “Presentation” is not limited to written work. It also includes oral presentations, computer assignments and artistic works. If you translate the work of another person into French or English and do not cite the source, this is also plagiarism. If you cite your own work without the correct citation, this too is plagiarism. In simple words:

DO NOT COPY, PARAPHRASE OR TRANSLATE ANYTHING FROM ANYWHERE WITHOUT SAYING FROM WHERE YOU GOT IT! DON’T FORGET TO USE QUOTATION MARKS!

Source: <http://provost.concordia.ca/academicintegrity/plagiarism>

**Access Centre for Students with Disabilities:** please visit  
<http://concordia.ca/students/accessibility>

**Student Success Centre:** please visit  
<http://concordia.ca/students/success>

## Tentative Class Schedule

The exact syllabus for the course will depend on how quickly we cover the introductory chapters. In general, our aim for this semester is to cover logic, functional, procedural, imperative, object-oriented, as well as aspect-oriented programming paradigms. In this course we cover many languages.

Part	Topics
<b>Intro</b> (C1)	<ul style="list-style-type: none"><li>• Introduction to the Course</li><li>• Object Oriented Programming using Java (part IV)</li><li>• Advanced Topics in Java</li></ul>
<b>LOGICAL</b> (C2 and C3)	<ul style="list-style-type: none"><li>• Logic Programming using PROLOG (part I)</li><li>• <b>Test 1</b></li><li>• Assignment 1</li></ul>
<b>FUNTIONAL</b> (C4, C5, and C6)	<ul style="list-style-type: none"><li>• Functional Programming using LISP (part II)</li><li>• <b>Test 2</b></li><li>• Assignment 2</li></ul>
<b>PROCEDURAL</b> (C7, C8, and C9)	<ul style="list-style-type: none"><li>• Procedural Programming using C (part III)</li><li>• <b>Test 3</b></li></ul>
<b>MULTI PARADIGM</b> (C10, and C11)	<ul style="list-style-type: none"><li>• Multi Paradigm Programming using Ruby (part VI)</li></ul>
<b>AOP</b> (C12 and C13)	<ul style="list-style-type: none"><li>• Aspect Oriented Programming using AspectJ (part V)</li><li>• Assignment 3</li><li>• <b>Test 4</b></li></ul>

## Important Dates

Assignment 1 due: Tuesday, Oct 6, 2020

Assignment 2 due: Tuesday, Nov 3, 2020

Assignment 3 due: Tuesday, Dec 8, 2020

**Test 1:** Friday, Oct 2, 2020 3PM-5PM (75-90 min timed exam)

**Test 2:** Friday, Oct 30, 2020 3PM-5PM (75-90 min timed exam)

**Test 3:** Friday, Nov 20, 2020 3PM-5PM (75-90 min timed exam)

**Test 4:** to be scheduled by the exam's office during the final exams period

**Please note: In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.**