CISC 360 Fall 2019:
Syllabus (PRELIMINARY)

Joshua Dunfield

September 5, 2019

1 Course introduction

CISC 360: Programming Paradigms introduces the two major "programming paradigms" that are not covered in other courses at Queen's. These two paradigms are *functional programming* and *logic programming*. Both feel very different from imperative programming and object-oriented programming, which are used in most of our other courses.

It is common to classify programming *languages* into these paradigms: Java "is" object-oriented (and, I would say, imperative), C "is" imperative, C++ is object-oriented and imperative, and Python is... well, people sometimes call it a scripting language, but it could also be called object-oriented, imperative, or functional.

When people say that Java is object-oriented, what they really mean is that Java makes the *style* of object-oriented programming *easier*: many of Java's language features, such as classes and inheritance, are fundamental to writing programs that "feel" object-oriented. But you can write Java programs that don't use inheritance.

To introduce you to the two new paradigms, we'll use two programming languages that support these paradigms:

- Haskell is said to be a functional programming language, because it supports a "functional style": the main building block is the *function* (not methods and classes), and functions are "pure"—inputs are fed into functions, which return outputs, which are used as the inputs to other functions. You can't reassign to a variable in Haskell (I'm oversimplifying, but that's the basic idea).

- Prolog is a logic programming language, because it supports a "logical style": the main building block is the *clause*. A clause is a logical rule for deriving knowledge from known facts. Rather than calling methods or functions, we interact with Prolog by making *queries*.You might ask a Prolog program "is the Queen's School of Kinesiology building in Kingston?". Based on the two facts that (1) the building is on the Queen's main campus, (2) the Queen's main campus is in Kingston, and a "transitive rule" that if A is in B and B is in C then A is in C, the program would answer "yes". You could also ask the Prolog program for all buildings known to be on the main campus, or all buildings known to be in Kingston.

This course requires you to learn two programming languages that are (probably) completely new to you. So the universal advice to students—**start early**—is especially applicable. For example, even if you can correctly estimate how long it will take you to do a programming assignment in Python or Java (and, therefore, procrastinate doing the assignment until shortly before the deadline), you probably can't estimate how long it will take you to do a Haskell or Prolog assignment. Different programming styles make different things easy or hard. At this point, you have no idea what will take half as long as you expect, and what will take twice as long as you expect. (I like functional programming because it makes most of the things *I* need to do easy. However, I am an academic programming languages researcher, which is not the common case.)

A benefit of having to learn two new languages, and two new paradigms, is that it will be easier to learn *more* new languages. If you keep on programming for more than a few years, you will probably have to learn several programming languages that don't exist yet. (When I was an undergrad, C++ was overwhelmingly popular in industry; Python, Java, and JavaScript had just been invented.)

1.1 What's changing from 260?

Two administrative changes were made: First, 260 is renumbered to 360. Second, 204 is now a pre-requisite instead of a pre- *or co-requisite*, so you can no longer take 204 at the same time as this course.

Because of the first change, we can assume more knowledge and experience generally, which allows us to move (somewhat) faster and cover the material in more depth.

Because of the second change, students in 360 have already received a C– or better in 204, which allows us to start building on 204 background material *immediately*. Previously, it would have been unfair to students taking 204 at the same time as this course to assume they had seen material they would not see until the end of the term. Now we can assume everyone has passed 204.

1.2 Some learning outcomes

After doing well in 360, students should be able to:

- Write short programs in a functional language such as Haskell or LISP, including the use of recursion, lists, higher-order functions.
- Use structural induction to prove simple assertions about functional programs.
- Write short programs in a logical language such as Prolog.
- Predict the behaviour of small programs written in either paradigm.

2 Logistics

Instructor: Joshua Dunfield
Email: joshuad at cs.queensu.ca (please put "360" in the subject line)
Office Hours: Goodwin 534, Wed. 13:00–14:00 + additional times TBA + by appointment
Note: I expect office hours to start on 2019–09–12 (also known as September 12th, 2018).

Lectures: Tue 12:30–13:30, Thu 11:30–12:30, Fri 13:30–14:30 in Kinesiology 100
TAs: Meara Donovan, Val Kobilaski
Attendance is neither mandatory nor tracked, but attending class is generally a good idea.

Textbooks: There are two required textbooks:

- Simon Thompson. *Haskell: The Craft of Functional Programming,* third edition.

- Ivan Bratko. *Prolog Programming for Artificial Intelligence,* fourth edition.

The Queen's bookstore sells these for a single package for around $200 plus tax. You may wish to obtain used copies (both books were used repeatedly in CISC 260).

Software: We will use the following:

- GHC (Glasgow Haskell Compiler)
- SWI-Prolog

You will also want a text editor, such as Notepad++ (on Windows), Atom, Emacs, or Aquamacs (macOS). (Despite its name, macOS TextEdit is not a suitable text editor.)

The School of Computing lab computers should have GHC, SWI-Prolog, and Notepad++ installed, but you will probably want to download your own copies.

3 Assessment

The basis for your course grade will be:

20%	Assignments
50%	Quizzes
30%	Final examination

I expect approximately six assignments, every 1–2 weeks. Since I usually drop the lowest assignment mark, this means an assignment is worth 4% of the course grade on average (some assignments may be worth more than others).

The percentages above are not the whole story. Read the following **two rules**; these are "good" (they can only raise your grade).

3.1 Rule: One-time forgiveness (assignments)

In general, I will drop your lowest assignment mark. This includes situations where you submitted nothing and received a zero: zero is a very low mark, so it would be dropped.

Exception: If a student receives a reduced assignment mark (such as zero) as a consequence of a departure from academic integrity on that assignment, that assignment mark will **not** be dropped—instead, all assignments will be counted when calculating the course grade (just as if this rule did not exist).

3.2 Rule: One-time forgiveness (quizzes)

In general, I will drop your lowest quiz mark. This includes situations where you missed a quiz entirely (for any reason) and received a zero: zero is a very low mark, so it would be dropped.

Exception: If a student receives a reduced quiz mark (such as zero) as a consequence of a departure from academic integrity on that quiz, that quiz mark will **not** be dropped—instead, all quizzes will be counted when calculating the course grade (just as if this rule did not exist).

4 TENTATIVE Quiz Dates (in class)

Before telling me that you can't attend on one of these dates, please read the rule described above (**??**, One-Time Forgiveness (quizzes)).

- Quiz #1: 2019–10–03 (Thursday)
- Quiz #2: 2019–10–24 (Thursday)
- Quiz #3: 2019–11–14 (Thursday)
- Quiz #4: 2019–11–28 (Thursday)

5 Additional syllabus information

5.1 Common syllabus

The School of Computing's "Common Syllabus Information (2019–2020)" is part of the 360 syllabus. Access it at the following link:

`http://www.cs.queensu.ca/students/undergraduate/syllabus/year2019-20.php`

5.2 Academic integrity

For greater certainty, an excerpt from the standard syllabus's section on "Academic Integrity" is reproduced here:

Queen's students, faculty, administrators and staff all have responsibilities for supporting and upholding the fundamental values of academic integrity. Academic integrity is constituted by the five core fundamental values of honesty, trust, fairness, respect and responsibility (see `www.academicintegrity.org`*) and by the quality of courage. These values and qualities are central to the building, nurturing and sustaining of an academic community in which all members of the community will thrive. Adherence to the values expressed through academic integrity forms a foundation for the "freedom of inquiry and exchange of ideas" essential to the intellectual life of the University.*

Students are responsible for familiarizing themselves with and adhering to the regulations concerning academic integrity. General information on academic integrity is available at Integrity@Queen's University, along with Faculty or School specific information. Departures from academic integrity include, but are not limited to, plagiarism, use of unauthorized materials, facilitation, forgery and falsification. Actions which contravene the regulation on academic integrity carry sanctions that can range from a warning, to loss of grades on an assignment, to failure of a course, to requirement to withdraw from the university.

For this instance of CISC 360: Assignments must be completed **individually**. You are encouraged to consult the TAs and the instructor, but you must not consult other students, copy solutions from the web, etc. This does not prevent you from studying course material together (which is not only allowed, but encouraged), but we are assessing your individual performance in completing each assignment.

If you violate this policy, you are not only departing from integrity, but are setting yourself up to fail the quizzes and final examination; your instructor likes exam questions that are closely related to the assignments.

Quizzes and exams must, of course, be done individually.