CSCI 235 - Programming Languages, Fall 2019

Hans de Nivelle, Selim Temizer

August 20, 2019

1 Read this Syllabus!

This syllabus contains important information about the course CSCI 235 (Programming Languages). Read it completely.

2 Time/Location/People

Lectures:

1L	12.00-13.15	Tuesday/Thursday
2L	13.30-14.45	Tuesday/Thursday

(All lectures take place in room 7e.429.)

Laboratories:

1Lb	09.00-10.50	Monday
2Lb	11.00-12.50	Monday
3Lb	15.00-16.50	Monday
4Lb	09.00-10.50	Wednesday
5Lb	11.00-12.50	Wednesday
6Lb	15.00-16.50	Wednesday

(All labs take place in room 7.422.)

Lecturer:	Hans (Jean) de Nivelle (hans.denivelle@nu.edu.kz)	(appointment by email)
Lecturer:	Selim Temizer (selim.temizer@nu.edu.kz)	(Tue/Thu 15.00-16.00)
Teaching Assistant:	Banu Suleimenova (banu.suleimenova@nu.edu.kz)	(appointment by email)
Teaching Assistant:	Daniyar Kultayev (daniyar.kultayev@nu.edu.kz)	(appointment by email)
Teaching Assistant:	Askhat Bukeyev askhat.bukeyev@nu.edu.kz	(appointment by email)

3 Prerequisites

The course is intended for 2nd year Computer Science students. You must have good experience with programming in an imperative programming language, such as C, $C^{\#}$, or C^{++}. You must have passed the prerequisite course CSCI 152 (Performance and Data Structures) with at least a C.

4 Course Description

The course covers a couple of important programming languages, namely Java, Python and Prolog. We try to teach you how to write good code in each of these languages, and hope that knowing about different types of programming languages will help you with choosing the right language for a given problem.

Using the three given languages as example, we will discuss important properties of programming languages: A language can be either *compiled* or *interpreted.* It can be *statically typed* or *dynamically typed.* It can be *imperative, functional*, or *logical.* An imperative language can have *value semantics*, or *reference semantics* (usually combined with object-orientation). A language can have built-in memory management (garbage collection), or leave memory management to the user.

- Java is a modern object-oriented programming language that is widely deployed on many different types of hardware around the world due to its high portability and support of a very large out-of-the-box class library. It is an interpreted language with optimized virtual machines that enable high-performance computation. Java has a simple syntax, yet it is an industrial-strength language.

- Python is interpreted, dynamically typed, imperative and it has reference semantics. It has built-in memory management. It supports OO. It has a big collection of libraries. It is slow but very flexible.

- Prolog is logical, interpreted, and it has built-in memory management. A program in Prolog is a logical description of the problem. The interpreter uses this description to search for solutions of the problem. Prolog is very suitable for search, and it is used in artificial intelligence.

In addition to teaching you the important details of the languages, we will try to teach you how to write good, efficient, and easily readable code in each of these languages.

5 Course Materials

The main material are the slides that will be used during the lectures. The slides will be regularly updated on Piazza, so that you have access to them. Students are expected to read these slides, and to regularly look for updates.

Students are not required to use text books, but the following text books are recommended if you are looking for additional material:

- Head First Java (A Brain Friendly Guide), Kathy Sierra and Bert Bates, O'Reilly, 2005.
- Java (How to Program), Paul Deitel and Harvey Deitel, Early Objects (11th Edition), Pearson Education Limited, 2018.
- Design Patters (Elements of Reusable Object-Oriented Software), Erich Gamma and Richard Helm and Ralph Johnson and John Vlissides, Pearson Education, 1995.
- Core Java (Volume 1), Cay S. Horstmann and Garry Cornell, Prentice Hall, 2015.
- Computing with Python (An introduction to Python for Science and Engineering), Claus Führer, Jan Erik Solem, Olivier Verdier, Pearson Eduction Limited, 2014.
- `https://www.python.org/`
- The Art of Prolog (2nd Edition): Advanced Programming Techniques, Leon Sterling, Ehud Shapiro, MIT Press Cambridge, 1994.

6 Course Organization

The course has two weekly lectures of 75 minutes each. You are expected to attend.

In addition to the lectures, there will be lab sessions, which last 110 minutes. You must have registered for one of the lab sessions, and you are obliged to attend, unless it is explicitly announced on Piazza that you don't have to. Different tasks will be checked in different ways. Some easy tasks will be checked at the end of the lab session. Some tasks may be checked one week later in the lab. Other tasks may be checked off-line which means that you have to submit them in Moodle. In order to to get full credit, your code must work, have good design, proper indentation, must be submitted on time, and fulfill possible other requirements that are specified in the task.

Late submissions will not get the full number of points, and very late submissions will not even be looked at. For details about grading, we refer to Section 9.

7 Academic Integrity

The government of Kazakhstan has installed Nazarbayev University, because it believes that a Western style university is essential for the economical, scientific and cultural development of Kazakhstan.

Two things are essential for the functioning of NU, **(1)** that students who graduated from NU are highly-qualified, independent workers whose judgement and honesty can be trusted without hesitation, and **(2)** that everyone, in Kazakhstan and in the rest of the world, knows about this.

Because of this, Nazarbayev University and the School of Engineering and Digital Sciences have established high standards for academic integrity, using an approach in which students are trained to produce original work according to professional standards, and to properly cite and reference the work of others when it is appropriate to do so.

Cheating is a direct threat to both of the goals mentioned above. We hope that you understand this and that you will not attempt it. In particular, don't do the following things:

- Don't hand in work that you did not make by yourself. It is possible to work together with a friend, it is allowed to ask for advice, but it is not OK to copy work.

- Don't give your work to others if there is a chance that they will copy it and hand it in as their own. Again, it is allowed to help your friends, but it is not OK to give them your solutions.

- Although exercises in the labs are usually graded, their true purpose is to teach you. Therefore, it is acceptable to present incomplete work, as long as you made a serious attempt to complete it and can explain what your difficulties are. We will try to answer your questions, and help you with making the task, as long as we believe that you seriously tried.

  For learning, it is much better to ask for help in the lab, than to copy work from somebody else, or from the internet. It is also possible to visit one of the TAs during office hours.

- Don't come begging for a better grade. It is allowed to have a look at the exam with the purpose of improving yourself. It is also reasonable to expect that true grading errors are corrected, but it is not OK to ask for better grade if you think that the exam was checked too strictly. We apply the same rules to everyone equally. If we give you a grade that you did not deserve, we are a threat to the two goals mentioned above, so we will not do that.

We are sure that nearly everyone understands and agrees with the arguments above, but since there are always a few who don't, we also have to inform you of the following:

- In the event that we discover academic misconduct such as plagiarism or other forms of cheating, the student will receive no credit for the work, and the event will be reported to the Dean of SEDS.

- Severe cases, or a repeated offense, may result in failure of the course and suspension or expulsion from the university.

- In case somebody writes a name in an attendance list of a person who is not present in the lecture, we will deduce points from the person who wrote the false name.

- Programming exercises may be tested by an automated plagiarism detector, with or without suspicion. Discovered plagiarism is likely to result in loss of points, possible with a report to the Dean of SEDS.

- In the case a student argues very much about grades for a part of an exam or quiz, we may decide to recheck the complete exam of the student.

8 Expected Behavior

8.1 In class

You are expected to attend lectures, to be attentive during these lectures, and to arrive on time. You are expected to read the discussed material soon after the lecture, and to try to read in advance for the next lecture. Relevant materials will be made available in Piazza.

8.2 During lab

Coding is the art of writing algorithms that can be efficiently run on a computer, and that can be read and understood by a person. Therefore, we believe that code should be checked in person as much as possible. This is difficult to organize with large groups of students, but we still try to inspect code manually. This implies that it is not sufficient to submit code 'that works'. Code must be maintainable and well-structured, and must meet well-defined quality criteria. Moreover, code that has bad style, usually doesn't work when carefully tested. Please cooperate with this system to make it work. In particular, be on time, and be well-prepared. Don't go into long arguments when your code is criticized. If possible, show your work on a lab computer. It is more efficient, and we have little time.

8.3 During Quiz Reviewing Sessions

Purpose of reviewing sessions is to allow the students to see what mistakes were made during the quiz, i.e. to have a last chance to understand the material before the lecture moves on to another topic. The purpose is not to argue about grades. Don't put unreasonable pressure on professors or TAs to change a grade. A grade is an objective measurement of how much the student learnt during the class. Giving in to pressure would be unfair towards your peers. We have procedures in place that make it impossible to TAs or professors to change a grade on the spot.

8.4 At all times

Do not copy work from other students, and do not allow other students to copy from your work. Giving advice and cooperation are fine, but copying is not fine. Regularly check the Piazza system for announcements and for updated material. Regulary read your emails.

9 Grading

Final grades will be calculated using the following weights:

Attendance and Participation	5 pts
Labs and Programming Assignments	20 pts
Quizzes (best 6 out of 7)	40 pts
Final Exam	35 pts

This results in a total of 100 pts. For calculation of the final letter grades, we use Nazarbayev University's standard cut off borders:

Grade	At Least	Less Than
A	95	
A^-	90	95
B^+	85	90
B	80	85
B^-	75	80
C^+	70	75
C	65	70
C^-	60	65
D^+	55	60
D	50	55
F		50

10 Grading of Late Submissions

Work that is handed in late without medical excuse (sprawka) will not receive the full credit. If the work is very late, it will not be graded at all.

11 Quiz and Exam Reviewing

A grade is an objective measurement of how well the student fulfilled the learning goals of the course. It is not a present, nor a reward for effort.

After the quizzes and the final exam, we will have reviewing sessions so that the student can review his/her work. Purpose of such sessions is to give the student a last chance to understand the covered material. If the student believes that a given grade does not properly represent what the student has learnt, the student may file a grade correction form. Grade correction forms are usually considered within a few days, and the student receives the decision by email.

Putting pressure on professors or TAs with the aim of improving a grade for other reasons is useless, because we have rigid procedures in place that don't allow a single person to change a grade.

12 Schedule

Last day to ADD: August 16, 2019 (Friday)

Last day to DROP: August 23, 2019 (Friday)

Last day to WITHDRAW: (with W grade): October 9, 2019 (Wednesday)

Week Number	Dates	Topics
1	August 12-18	Introduction; Compiled vs. Interpreted Languages Java Basics
2	August 19-25	Object-Oriented Programming; Inheritance, encapsulation, polymorphism
3	August 26 - September 1	Unified Modeling Language (UML); Generics; Packages; Collections
4	September 2-8	Exception Handling Graphical User Interface (GUI); Swing
5	September 9-15	Streams; Multi Threading
6	September 16-22	Examples of Software Design Patterns; Object-Oriented Design Principles (SOLID)
7	September 23-29	Java Virtual Machine (JVM) Internals; Java Security
	September 30 - October 6	**Fall Break**
8	October 7-13	Introduction to Python: An interpreted language with reference semantics and complete absence of static typing. Differences between Python 2 and Python 3.
9	October 14-20	Container Types: Tuples, Dictionaries, Sets, Iterators. Dangers of Reference Semantics
10	October 21-27	Object-Oriented Design in Python Member functions, dynamic type checking
11	October 28 - November 3	**SciPy** and **NumPy** libraries, Matrices and Arrays Plotting
12	November 4-10	Introduction to Prolog. Prolog's Search Algorithm on Propositional Logic
13	November 11-17	Prolog's Search Algorithm with Unification Prolog with Data: Lists, Terms
14	November 18-24	Using Prolog for search, use of cut Negation as failure
	November 25 - December 4	Final Exam Period. There will be a final exam over the complete material.

(Weeks 2,4,6,8,10,12,14 have quizzes, always on Thursday, unless announced different in Piazza.)