EXPANDED COURSE DESCRIPTION
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
Lassonde School of Engineering
Electrical Engineering Computer Science
LE / EECS 1021 3.0 SECTION E
OBJECT ORIENTED PROGRAMMING
SUMMER 2019

COURSE CALENDAR DESCRIPTION

"Introduces student to computational thinking - a process-based approach to problem solving. It uses a problem-based pedagogy to expose the underlying concepts and an experiential laboratory to implement them. The programming language is chosen so that it is widely used in a variety of applications, is object-oriented, and is of industrial strength (Java is an example of such a language). The problems are chosen in order to expose abstract programming concepts by immersing them in relevant and engaging applications. The experiential laboratory is based on sensors and actuators that connect to a computer. The problems are chosen with consultation with the various engineering disciplines in the Faculty with a view of exposing how computing is used in these disciplines. Prerequisites: LE/EECS1011 3.00. Course credit exclusions: LE/EECS 1022 3.00. Previously offered as: LE/EECS1020 3.00, LE/CSE 1020 3.00.

The objective of 1021 is to introduce computational thinking - a process-based approach to problem solving. It uses a problem-based pedagogy to expose the underlying concepts and an experiential laboratory to implement them. The programming language is chosen so that it is widely used in a variety of applications, is object-oriented, and is of industrial strength (Java is an example of such a language). The problems are chosen in order to expose abstract programming concepts by immersing them in relevant and engaging applications. The experiential laboratory is based on sensors and actuators that connect to a computer. By identifying sensors as concrete inputs and actuators as output objects, students can build event-driven software that connects events to actions. The problems are chosen with consultation with the various engineering disciplines in the Faculty with a view of exposing how computing is used in these disciplines. The lab hardware is chosen so that it can interface with a variety of languages (including MATLAB and Java); has several analog-to-digital and digital-to-analog converters; can control external power supply, and has a form factor suitable for undergraduate labs. Learning Objectives for the Course: By the end of the course, the students will be able to: 1. Demonstrate the ability to test and debug a given program and reason about its correctness. 2. Given a problem specification and a suitable API, build an application that meets the given requirement. 3. Use ready-made collections to solve problems involving aggregations of typed data. 4. Build an event-driven application that controls sensors and actuators in order to connect events to physical actions. 5. Program common applications from a variety of engineering disciplines using an object oriented language and solve them on the computer.

INSTRUCTOR(S)

<table>
<thead>
<tr>
<th>Name</th>
<th>Section / Format / Term</th>
<th>Contact Email</th>
<th>Contact Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mani, Reza</td>
<td>Sec. E / LECT / SU</td>
<td><a href="mailto:rmani@yorku.ca">rmani@yorku.ca</a></td>
<td></td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

COURSE LEARNING OUTCOMES:

By the end of the course, the students will be able to:
• Demonstrate the ability to test and debug a given program and reason about its correctness.
• Given a problem specification and a suitable API, build an application that meets the given requirement.
• Build an event-driven application that controls sensors and actuators in order to connect events to physical actions.
• Program common applications from a variety of engineering disciplines using an object oriented language and solve them on the computer

RECOMMENDED TEXTBOOKS:
• Head First Java by Sierra and Bates [ebook from the YorkU Library]
• Beginning Java Programming: The Object-Oriented Approach by Baesens, Backiel, and vanden Broucke [ebook from the YorkU Library]
• Java: A Beginner's Guide by Schildt [available in Steacie Library]
• Absolute Java by Savitch [available in Steacie Library]

The main resources for the course will be the power points presented in the class (and posted on Moodle) and a number of instructive youtube tutorials. The link for the youtube tutorials will be given in the first lecture.

MARKING SCHEME:
Each piece of work in the course will be assigned a numerical grade. Individual grades will be combined based on the weightings given below. Your final numerical grade out of 100 will be converted to a university letter grade using the standard table for the mapping. There are no make-up tests, alternate mechanisms of evaluation, etc. Should you miss an evaluation due to medical reasons, a properly completed Attending Physician's Statement is required. Once available, marks will be posted on Moodle. Check marks as they appear.

The grade components of the course are as follows:
• Midterm (15%). The midterm will have a multiple choice part (50%) and a descriptive part (50%). The descriptive part will contain writing the outcome of a program and debugging a program. It will take place in mid-June. Exact date will be announced later.
• Labs (20%). There will be 7 labs in total. All the labs except lab 0 have an assignment completion part for 1% and a hardware coding and performance component for 2% (3% total). Lab 0 has two parts, each worth 1% and it has to do with installing the java software (Eclipse) and running one java program on you on your laptop to make sure it is running with no issues (part 1 worth 1%) and building a few console applications (The tasks will be given in the lab) using Java (part 2 worth 1%).
• Lab tests (30%). Two lab tests at 15% each - 30% total. These are two labs that you will conduct on your own. These are supervised events during which you will solve coding problems in the lab. The first lab test is around beginning of June (before the midterm) and second lab test is at the end of July.
• Final (35%). The final will be in multiple choice format and may also require a handwritten programming section. It will take place during the examination period. The exact date will be announced by the university prior to examination period.

ACADEMIC INTEGRITY LINKS
• Senate Policy on Academic Honesty - http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/
• Academic Integrity - http://lassonde.yorku.ca/academic-integrity

STUDENT LINKS
• Student Rights and Responsibilities - http://oscr.students.uit.yorku.ca/student-conduct
• Religious Observance - https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/regobs
• Academic Accommodation for Students with Disabilities - http://secretariat-policies.info.yorku.ca/policies/academic-accommodation-for-students-with-disabilities-policy/
• Student Accessibility Services (SAS) - https://accessibility.students.yorku.ca/
• York University’s Policies on Sexual Violence - http://secretariat-policies.info.yorku.ca/policies/sexual-violence-policy-on/
• York University’s Policies on Gender/LGBTQ+/Positive Space - http://rights.info.yorku.ca/lgbtq/

LAND ACKNOWLEDGEMENT
• We acknowledge our presence on the traditional territory of many Indigenous Nations. The area known as Tkaronto has been care taken by the Anishinabek Nation, the Haudenosaunee Confederacy, the Huron-Wendat, and the Métis. It is now home to many Indigenous Peoples. We acknowledge the current treaty holders, the Mississaugas of the New Credit First Nation. This territory is subject of the Dish With One Spoon Wampum Belt Covenant, an agreement to peaceably share and care for the Great Lakes region.

• The Indigenous Framework for York University: A Guide to Action can be found here: http://indigenous.info.yorku.ca/

• Meaning of a land acknowledgement: http://healthydebate.ca/opinions/indigenous-land-acknowledgements

Many courses utilize Moodle, York University’s course website system. If your course is using Moodle, click here to access it.

Moodle @ York University