Instructor Information and Office hours

Dr. Mokhtar Beldjehem, Room ME4239, MokhtarBeldjehem@cunet.carleton.ca, Office hours are posted in cuLearn.

TA Information and Office hours

TAs will be available in the labs. A schedule will be posted in cuLearn.

Calendar Description

SYSC 2004: Object-Oriented Software Development

Calendar description: http://calendar.carleton.ca/undergrad/courses/SYSC/

Prerequisites

The prerequisites for SYSC 2004 are: SYSC 2006 or permission of the department, and second-year status in Engineering

Assumed Knowledge

Students should:

- understand the concepts that underlie most imperative programming languages and be able to use this knowledge to help them learn new languages;

- understand how memory is managed by an executing program, and demonstrate this knowledge pictorially;

- understand different designs for simple abstract linear collections such as lists (vectors), queues and stacks;

- be able to construct simple recursive functions;

- and be prepared to undertake this course that provides a thorough introduction to object-oriented programming principles

Students who have not satisfied the prerequisites for this course must either withdraw from the course or obtain a prerequisite waiver by visiting the Engineering Undergraduate Academic Support Office.
Course Objectives

Designing and implementing small-scale programs as communities of collaborating objects, using a dynamically-typed or statically-typed programming language. Fundamental concepts: classes, objects, encapsulation, information hiding, inheritance, polymorphism. Iterative, incremental development and test-driven development.

Learning Outcomes

By the end of this course, students should be able to:

1. Identify the fundamental concepts of object-oriented programming (classes, objects, encapsulation, information hiding, inheritance, polymorphism).
2. To implement small-scale programs as communities of interacting (collaborating) objects
3. To apply lightweight, modern techniques commonly used during object-oriented software development (iterative, incremental development; test-driven development).
4. To draw basic class and object diagrams.
5. To apply their object-oriented knowledge to developing simple graphical user interfaces.

Graduate Attributes (GA’s)

The Canadian Engineering Accreditation Board requires graduates of engineering programs to possess 12 attributes at the time of graduation. Activities related to the learning outcomes listed above are measured throughout the course and are part of the department’s continual improvement process. Graduate attribute measurements will not be taken into consideration in determining a student’s grade in the course. For more information, please visit: https://engineerscanada.ca/.

<table>
<thead>
<tr>
<th>Graduate Attribute</th>
<th>Learning Outcome(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.S: Knowledge Base: Developed: Programming and algorithms</td>
<td>2, 5</td>
</tr>
<tr>
<td>1.8.S: Knowledge Base: Introductory: Software engineering</td>
<td>1</td>
</tr>
<tr>
<td>3.1: Investigation: Introductory: Complex problem assessment</td>
<td>3, 5</td>
</tr>
</tbody>
</table>
Textbooks (or other resources)


Printed copies can be purchased at the university bookstore. An online version of the eTextbook can be rented for 180 days from CourseSmart (www.coursesmart.com) for roughly 40% of the list price of a printed copy.

The Third, Fourth or Fifth Editions of this book are also acceptable.


For students who like zyBooks, there is a "Programming in Java with zyLabs" available at zyBooks.com. However, I recommend the course text book instead.

**Evaluation and Grading Scheme**

Students will be evaluated by means of laboratory work, assignments, a midterm exam and a final exam.

To pass the course, students must pass the final examination (50% or better). For students who pass the final exam, a numeric mark out of 100 will be calculated by weighting the course components as shown here:

- Class participation up to 2% bonus (subject to availability of TurningPoint clickers)
- Lab work 10% (up to 2% bonus available)
- Assignments 15%
• Midterm exam 25%

• Final Exam 50%

Breakdown of course requirements

Lab Periods

Attendance at the scheduled laboratory periods is mandatory, and attendance will be taken. During the labs you will work on short programming exercises that are intended to help you understand particular concepts that have been introduced in the lectures. You will normally be required to demonstrate and/or submit your lab work by the end of the lab period, as indicated in that week’s lab "handout”. Your work in each lab period will be given a mark of 0, 0.5, or 1.

There will be twelve labs. This means you can earn up to 12/10 (i.e. up to 2 bonus marks) for the lab component of the course.

If you are absent from a lab period for any reason, you will receive 0/1 for that lab. If you are unable to attend a lab because of illness, you may attend a different lab section (for the same lab) to get your work checked. If you cannot attend another lab, you should do the missed lab work on your own time; however, you cannot submit your completed lab work late to receive credit for the missed lab.

Assignments

There will be five programming assignments. Your lowest assignment mark will not be counted when calculating your final grade. This means you can miss an assignment and still earn full marks (100%) for the assignment component of the course. Late assignments will normally not be accepted. In addition, there will be a bonus sixth assignment. Assignments are to be done on your own. Collaboration or sharing code is considered plagiarism.

Exams

There will be one closed-book midterm exam, which will be held during the lecture on Fri. Feb. 14th for Section C and on Thu. Feb. 13th for Section D.

Students who are unable to write the midterm exam because of illness or other circumstances beyond their control must provide in cases of illness a medical certificate dated no later than one working day after the exam, or appropriate documents in other cases. If this information is provided to the instructor no later than three working days after the exam, the student will be eligible for a make-up midterm; otherwise, the mark for the missed midterm exam will be 0.
A closed-book final exam will be held during the University's final examination period. All students are eligible to write the final examination, regardless of the marks they received during the term. The final examination is for evaluation purposes only and will not be returned to students. You will be able to make arrangements with the instructor or with the department office to see your marked final examination after the final grades have been made available.

**Tentative Week-by-Week breakdown**

The order in which topics are presented may be changed as the course progresses.

- **Lectures 1 and 2:** Introduction to object-oriented concepts. Using Java classes and objects; introduction to the BlueJ Integrated Development Environment. (Chapter 1)

- **Lectures 3 and 4:** Java class definitions: fields (instance variables), constructors, instance methods. Accessor and mutator methods. (Chapter 2). UML representation of classes and objects.

- **Lectures 5 and 6:** Classes as types. The new operator. References to objects. Interacting objects. (Chapter 3)

- **Lectures 7 and 8:** Grouping objects: array objects. (Chapter 7) Generic collections: class ArrayList and LinkedList. (Chapter 4).

- **Lectures 9 and 10:** Using classes from Java's library (class String, wrapper classes) (Chapter 6), packages, reading and writing documentation (javadoc) comments (Chapter 8); Unit testing (JUnit) and debugging. (Chapter 9)

- **Lectures 11 and 12:** Inheritance: fundamental concepts, UML notation; support for inheritance in Java: the extends reserved word, constructor chaining, visibility of inherited variables. (Chapter 10).

- **Lecture 13:** Midterm exam

- **Lecture 14:** Inheritance, continued: subclasses and subtypes, polymorphic variables, up/ downcasting, class Object. (Chapter 10)

- **Lectures 15 and 16:** Inheritance, continued: method overriding, dynamic method lookup, method polymorphism, the super call. Overriding methods inherited from Object (equals, toString). (Chapter 11) Handling errors with exceptions, the exception class hierarchy (Chapter 14).

- **Lectures 17 and 18:** Inheritance, continued: abstract classes, interfaces (Chapter 12). Comparison of inheritance and composition as techniques for object-oriented design.
• **Lectures 19 and 20:** Introduction to design patterns. The observer pattern and event handling. (Chapter 13)

• **Lectures 21, 22, and 23:** Graphical user interface programming: the Swing framework, GUI components and event handling. (Chapter 13)

• **Lectures 24 and 25:** Review

**General Regulations**

**Attendance:** Students are expected to attend all lectures and lab periods. The University requires students to have a conflict-free timetable. For more information, see the current *Undergraduate Calendar, Academic Regulations of the University, Section 2.1.3, Course Selection and Registration* and *Section 2.1.7, Deregistration.*

**Health and Safety:** Every student should have a copy of our Health and Safety Manual. A PDF copy of this manual is available online: [http://sce.carleton.ca/courses/health-and-safety.pdf](http://sce.carleton.ca/courses/health-and-safety.pdf)

**Deferred Term Work:** Students who claim illness, injury or other extraordinary circumstances beyond their control as a reason for missed term work are held responsible for immediately informing the instructor concerned and for making alternate arrangements with the instructor and in all cases this must occur no later than three (3.0) working days after the term work was due. The alternate arrangement must be made before the last day of classes in the term as published in the academic schedule. For more information, see the current *Undergraduate Calendar, Academic Regulations of the University, Section 4.4, Deferred Term Work.*

**Appeal of Grades:** The processes for dealing with questions or concerns regarding grades assigned during the term and final grades is described in the *Undergraduate Calendar, Academic Regulations of the University, Section 3.3.4, Informal Appeal of Grade* and *Section 3.3.5 Formal Appeal of Grade.*

**Academic Integrity:** Students should be aware of their obligations with regards to academic integrity. Please review the information about academic integrity at: [https://carleton.ca/registrar/academic-integrity/](https://carleton.ca/registrar/academic-integrity/). This site also contains a link to the complete Academic Integrity Policy that was approved by the University's Senate.

**Plagiarism:** Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated.

Academic Accommodation: You may need special arrangements to meet your academic obligations during the term. You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at
For an accommodation request, the processes are as follows:

- **Pregnancy or Religious obligation:** Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details see [https://carleton.ca/equity/wp-content/uploads/Student-Guide-to-Academic-Accommodation.pdf](https://carleton.ca/equity/wp-content/uploads/Student-Guide-to-Academic-Accommodation.pdf)

- **Academic Accommodations for Students with Disabilities:** The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your *Letter of Accommodation* at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (*if applicable*). Requests made within two weeks will be reviewed on a case-by-case basis. After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website ([www.carleton.ca/pmc](http://www.carleton.ca/pmc)) for the deadline to request accommodations for the formally-scheduled exam (*if applicable*).

- **Survivors of Sexual Violence:** As a community, Carleton University is committed to maintaining a positive learning, working and living environment where sexual violence will not be tolerated, and where survivors are supported through academic accommodations as per Carleton’s Sexual Violence Policy. For more information about the services available at the university and to obtain information about sexual violence and/or support, visit: [https://carleton.ca/sexual-violence-support/](https://carleton.ca/sexual-violence-support/).

- **Accommodation for Student Activities:** Carleton University recognizes the substantial benefits, both to the individual student and for the university, that result from a student participating in activities beyond the classroom experience. Reasonable accommodation must be provided to students who compete or perform at the national or international level. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, see [https://carleton.ca/senate/wp-content/uploads/Accommodation-for-Student-Activities-1.pdf](https://carleton.ca/senate/wp-content/uploads/Accommodation-for-Student-Activities-1.pdf)

**Copyright on Course Materials:** The materials created for this course (including the course outline and any slides, posted notes, labs, project, assignments, quizzes, exams and solutions) are intended for personal use and may not be reproduced or redistributed or posted on any web site without prior written permission from the author(s).