Carleton University
Department of Systems and Computer Engineering

SYSC 2004 A **Object-Oriented Software Development** **Early Summer 2016**

Course Outline

Instructor: Dr. Lynn Marshall, Room ME4230, lynnmar@sce.carleton.ca

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).

Prerequisite

SYSC 2006 is the prerequisite for SYSC 2004. Prerequisite waivers will not normally be granted. Students who have not received credit for SYSC 2006 must withdraw from SYSC 2004 by the last date for registration; otherwise, they will be deregistered before the end of term.

Textbook

Objects First with Java: A Practical Introduction Using BlueJ, Fifth Edition, David J. Barnes, and Michael Kölling, Pearson/Prentice Hall, 2012, ISBN-13: 9780132492669.

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 or Fourth Editions of this book are also acceptable.

The URL for the book's Web site is http://www.bluej.org/objects-first.

Course Web Site

http://sce.carleton.ca/courses/sysc-2004/s16. Students are expected to access this site regularly, and consult it before emailing questions to the instructor. Portions of the Web site will be protected by a password, which will be provided to students in class. Please do not make this password publicly available.

Policy on Laptop and Tablet Computers

During scheduled labs, students who prefer to use their own laptop computers instead of the lab computers are permitted to do so; however, computers may be used only for course related work during the lectures.

Attendance

Students are expected to attend all lectures and lab periods. The University requires students to have a conflict-free timetable. Requests to accommodate a missed midterm exam, lab periods, due dates, etc., because of conflicts with jobs or vacation plans will not be considered.

Subject to availability, this course will use TurningPoint Clickers that can be signed out at the CUOL free of charge (details on web site). Students will receive up to 4% in bonus marks for getting a clicker and participating in clicker quizzes in the lectures.

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 4% bonus (subject to availability of TurningPoint clickers)
- Lab work 10% (up to 3% bonus available)
- Assignments 15%
- Midterm exam 25%
- Final Exam 50%

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 thirteen labs. This means you can earn up to 13/10 (i.e. up to 3 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 are not required to provide a medical certificate to explain your absence. It is up to you to 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. .

Exams

There will be one closed-book midterm exam, which will be held during part of the lecture on Thu May 26th.

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 five working days after the exam, the weight of the final exam will be increased to cover the missed midterm; otherwise, the mark for the missed midterm exam will be 0.

A closed-book final exam will be held during the University's June examination period. The *Academic Regulations of the University* permit instructors to specify requirements that must be satisfied for students to be eligible to write the final examination or, where circumstances warrant, the deferred final examination.

- All students are eligible to write the final examination, regardless of the marks they received during the term.

- Students who miss the final exam, but earned at least 60% overall on the lab and assignment component and wrote the midterm exam (or provided acceptable documentation to explain their absence from the exam), will receive the grade ABS. These students will be deemed to have performed satisfactorily during the term when their applications for a deferral of the final examination are considered. For more information, see the current Undergraduate Calendar, *Academic Regulations of the University*, Section 2.2, The Course Outline; Section 2.3, Standing in Courses/Grading System; and Section 2.5, Deferred Final Examinations.
- Students who miss the final exam but have not satisfied the conditions for receiving ABS, as listed above, will receive the grade FND. These students are ineligible to write the deferred final exam.

The final examination is for evaluation purposes only and will not be returned to students.

Students with Disabilities

Students with disabilities who require academic accommodations in this course are encouraged to contact a coordinator at the Paul Menton Centre for Students with Disabilities to complete the necessary letters of accommodation by June 3rd, 2016 for Early Summer Term (June exams). After registering with the PMC, make an appointment to discuss your needs with your instructor at least one week prior to the midterm exam.

Health and Safety

Every student should have a copy of our Health and Safety Manual. An electronic version of the manual can be found at http://www.sce.carleton.ca/courses/health-and-safety.pdf.

Week-by-Week Outline

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

- **Lecture 1:** Introduction to object-oriented concepts. Using Java classes and objects; introduction to the BlueJ Integrated Development Environment. (Chapter 1)
- **Lecture 2:** Java class definitions: fields (instance variables), constructors, instance methods. Accessor and mutator methods. (Chapter 2). UML representation of classes and objects.
- **Lecture 3:** Classes as types. The new operator. References to objects. Interacting objects. (Chapter 3)
- **Lecture 4:** Grouping objects: array objects. (Chapter 4) Generic collections: class ArrayList (Chapter 4) and HashMap. (Chapter 5).
- **Lecture 5:** Using classes from Java's library (class String, wrapper classes), packages, reading and writing documentation (javadoc) comments. (Chapter 5) Unit testing (JUnit) and debugging. (Chapter 7) Class variables and methods (the static reserved word). (Chapter 5)
- **Lecture 6:** Inheritance: fundamental concepts, UML notation; support for inheritance in Java: the extends reserved word, constructor chaining, visibility of inherited variables. (Chapter 8).
- **Lecture 7:** Inheritance, continued: subclasses and subtypes, polymorphic variables, up/ downcasting, class Object. (Chapter 8)
- **Lecture 8:** Inheritance, continued: method overriding, dynamic method lookup, method polymorphism, the super call. Overriding methods inherited from Object (equals, toString). (Chapter 9) Handling errors with exceptions, the exception class hierarchy (Chapter 12).
- **Lecture 9:** Inheritance, continued: abstract classes, interfaces (Chapter 10). Comparison of inheritance and composition as techniques for object-oriented design.
- **Lecture 10:** Introduction to design patterns. The observer pattern and event handling. (Chapter 13)
- **Lectures 11, 12, 13:** Graphical user interface programming: the Swing framework, GUI components and event handling. (Chapter 11)