

COMP 401

Foundations of Programming

General Course Info

Term:	Fall 2019
Department:	COMP
Course Number:	401
Section Number:	001
Credit Hours:	4
Time:	Tu/Th, 2:00-3:15
Location:	Genome Science G100

Instructor Info

Name:	Prof. Ketan Mayer-Patel
Office:	154 Fred Brooks Building
Email:	kmp@cs.unc.edu
Phone:	919-590-6009
Office Hours:	M / W, 8:30 – 10:00

Email Instructions

When sending email to Prof. Mayer-Patel about any class related issues, please be sure to do the following:

- Begin the subject line with the text “[COMP 401]”
- Include the following information at the end of your email:
 - Full Name (as officially registered)
 - PID

Learning Assistants (LAs)

To Be Announced

Textbooks and Resources

There is no required textbook for this course. Students interested in a text that covers the material in the course may find one of the following resources helpful:

- Thinking in Java, 3rd Edition, freely available here:
<http://www.cs.unc.edu/~kmp/comp401/TIJ/TIJ3.htm>
 - Note: there is a 4th edition that is sold via Amazon
- Head First Java, by Kathy Sierra

- Effective Java by Joshua Bloch
- Any other basic Java programming text.

Additionally, I will be posting links to various web-based resources.

Sakai

We will be using Sakai only for all official course communication (assignments, announcements, lecture notes, gradebook, etc.). Registered students should find that they already have access to the course Sakai site.

Piazza

Piazza will be used for peer-to-peer communication. Students should think of Piazza as a virtual study group attended by all the students in the class. In general, students should NOT expect Piazza to be a tool for communicating with instructors. Although instructional staff may answer questions and/or clarify answers provided by others, students should not necessarily expect an answer. Students should not post solutions or significant portions of programming assignments to Piazza. A more complete guide to appropriate Piazza use can be found here:

<http://www.cs.unc.edu/~kmp/comp401fall19/piazza-guide.html>

GitHub Classroom

GitHub Classroom will be used to distribute example code and material to students as well as for submitting assignment code. All students are expected to create a github account at <http://github.com>.

Course Description

This course is intended for people who already have some experience with programming. My goal is to teach you how to program well in an object-oriented style. Object-oriented programming emphasizes a modular design that promotes code reuse and structured solutions for complex problems that are easier to change and maintain. Almost all modern programming languages include some sort of object-oriented programming model.

We assume you have already learned the following basic programming concepts either in a formal course or via prior experience:

- primitive types (integers, real numbers, booleans)
- variables

- constants
- expressions
- assignments
- comments
- arrays
- loops
- procedures/functions/methods.

These concepts are taught in most if not all introductory programming courses regardless of whether they teach conventional or object-oriented programming. This course will teach you the next-level programming concepts. These include:

- objects
- classes
- interfaces
- polymorphism
- encapsulation
- abstraction
- inheritance
- delegation
- design patterns
- exceptions

Developing the skills that will enable you to use these concepts will form a large part of the challenge you face in this course. After this course, you will have a much deeper understanding of programming and learn some of the ideas that can make programming a science. We will be using Java as a vehicle for learning these concepts, but our ultimate aim is to provide you with a fundamental understanding of object-oriented programming that can be brought to bear to solve problems using the object-oriented features of any modern programming language.

Target Audience

The target audience for this course include students intending to major in Computer Science and students with some programming experience interested in developing a more formal approach to computer programming.

Prerequisites

COMP 110 or COMP 116 or an equivalent course in basic programming in high school or another institution. Students with sufficient experience with programming as a hobbyist or in a work environment may also take the course.

Course Requirements

Students will be required to complete 10 programming assignments, as well as attend mandatory recitations during which they will complete LA-led exercises. Quizzes given in lecture are also incorporated as part of the course grade.

Additionally, there will be 1 midterm and a final exam.

Key Dates

The following due dates for assignments and exam dates are subject to change.

Assignments:

Students will be expected to complete assignments in a timely manner as the semester progresses. Students are responsible for keeping up with assignment information and announcements as posted on Sakai, especially with regard to assignment deadlines.

Exams:

Midterm – Tuesday, 10/15, time and location TBA

Final – Saturday, 12/7, 12:00 pm – 3:00 pm, location TBA

Grading Criteria

Course grades are calculated as a weighted sum of the following:

- quizzes (10%),
- assignments (25%),
- midterm (30%),
- cumulative final exam (35%)

In addition, you must attend at least 9 out of 12 mandatory recitations to pass the class.

Course Policies

Assignments will be turned in using an on-line submission system that will be explained in class. Assignments are due by 6:00 AM of the due date. Late assignment submissions are subject to a late penalty of up to 50%. Assignments more than 3 days late will not be given credit.

The course final is given in compliance with UNC final exam regulations and according to the UNC Final Exam calendar.

Honor Code

All recitation exercises, quizzes, and exams must be completed individually and without external assistance or resources unless specifically authorized by the instructor or within the instructions of the exercise, quiz, or exam.

Students are expected to individually develop solutions for assignments. Students are allowed to help each other in the process of debugging programs and developing approaches to their solutions, but should not directly write code for or share code with each other. Students are expected to self-report any significant collaborations. Every student, however, is expected to be able to explain and discuss their code individually and may be asked to do so.

Students are encouraged to use the Piazza system in order ask and answer conceptual questions and/or seek a better understanding of the course material.

Exams will be closed book.

More information can be found here:

<http://www.cs.unc.edu/Admin/Courses/HonorCode.html>

Disclaimer

The professor reserves to right to make changes to the syllabus, including due dates and test dates. These changes will be announced as early as possible.