

Course Information

Course: COS 161 Algorithms in Programming
Call #: 40080
Credit Hours: 4
Room: Payson Smith 201
Day/Times: T/R 3:30 P.M.– 4:45 P.M.
Lab: T 5:00 P.M.– 5:50 P.M.in Luther Bonney 203
Prerequisites: Grade of C or higher in COS 160.
Textbook: Reges, S. & Stepp, M. (2019). *Building Java Programs A Back to Basics Approach* (5th ed.). Pearson. ISBN: 978-0135471944.

Instructor Information

Name: James Quinlan, Ph.D.
Office: 228 ScienceBuilding
Phone: (207) 780 – 4499
Email: james.quinlan@maine.edu
Office Hours: T/R 11:00 A.M.- 12:00 P.M.

Course description

The development of algorithms and their implementations in a higher-level programming language, with emphasis on proper design principles and advanced programming concepts. Introduction to the performance analysis of algorithms. Course requirements include a substantial number of programming projects.

Topics

- Classes
 - Inheritance and Interfaces
 - ArrayLists
 - Java Collections
 - Recursion
-

Learning Outcomes

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

- write Java classes using appropriate mechanisms to use information hiding, inheritance, and exceptions.
- explain the basic classes in the Java Collections Framework and know how to choose the appropriate one to solve a given problem.
- write small recursive programs.

- perform basic run-time analysis on code and will understand what it means for an algorithm to be linear, quadratic, etc.
 - articulate some of the tradeoffs exhibited when multiple solutions to the same problem are studied.
-

Additional Resources

- Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.
 - Sierra, K., & Bates, B. (2005). *Head First Java: A Brain-Friendly Guide*. O'Reilly Media, Inc.
 - Arnold, K., Gosling, J., & Holmes, D. (2005). *The Java programming language*. Addison Wesley Professional.
 - Bloch, J. (2008). *Effective Java (the Java series)*. Prentice Hall PTR.
 - Mrkvicka, J. (2019). *97 Things Every Programmer Should Know*.
-

Technology Requirements / Equipment

Typical technical suggestions for users are:

- **Java Code Development:** Students will write Java code using the Eclipse software, which will be pre-installed on lab computers. Students working from other locations must have sufficient storage and memory to install Eclipse and the associated Java software independently on their devices. Instructions for downloading and installing the software can be found at: <https://www.eclipse.org/downloads/packages/installer> Setting up Eclipse on Home Computer. (Eclipse software is distributed for free.)
- **External Storage:** Students completing work in the lab or pairs on another student's device should have access to an external storage mechanism. While a USB drive is recommended as a cost-effective and straightforward option, students opting for alternative long-term storage methods (e.g., cloud storage) are welcome to do so, provided they do not require technical support. Even the smallest USB drives available today should offer more than sufficient storage for COS 161. Students exclusively working off USB drives are encouraged to have a second drive as a backup.
- **.jar Files:** Students may be required to download .jar files occasionally for laboratory or assignment use. These files are small in size and freely distributed. A .jar (Java Archive) file is a package file format typically used to aggregate Java class files, resources, and metadata into a single file for distribution or deployment. A .jar file can contain compiled Java classes, resources (such as images or configuration files), and metadata necessary for executing a Java application. Similar to ZIP files, .jar files can be compressed to reduce their size, making them more efficient for distribution and download. Every .jar file has a manifest file (META-INF/MANIFEST.MF) that provides metadata about the contents of the .jar file. It includes information such as the main class to be executed when the .jar file is run. Some .jar files are executable and can be run directly from the command line or by double-clicking, provided they contain the necessary manifest information. .jar files are commonly used to package libraries or components that can be easily added to the classpath of a Java application. This simplifies the distribution and management of dependencies. The Java Development Kit (JDK) includes a utility called jar that can be used to create, view, and extract contents from

.jar files. Developers commonly use this tool during the development and deployment of Java applications. The same Java infrastructure supporting Eclipse will also support these files.

- **CodingBat Registration:** Students are expected to register for free accounts at the CodingBat website during the first laboratory exercise. CodingBat is accessible from almost any web browser, and both registration and use are free.

Course Evaluation and Grading Policies

Attendance and Participation (10%)

Regular attendance and fully engaged participation is expected of all students.

Quizzes (15%)

Regular quizzes will be given either in or outside of class on Brightspace. The dates for these quizzes will be announced during the lecture and through BrightSpace.

Laboratory assignments (24%)

Laboratory assignments will consist of a series of smaller tasks. These tasks may involve coding or analyzing Java programs. A given lab exercise may be done individually or as part of a pair; pairs may be varied throughout the semester, but a given week's assignment must be completed as part of a single pair (no swapping within a week) or individually. Only one assignment (with both students' names) must be turned in when a pair completes work. Both members are responsible for everything written in the assignment and will receive the same grade. If schedules require a pair to "divorce" after beginning the assignment but before turning it in, each must complete it individually. Lab assignments will generally be due at 4 p.m. on the Friday following the lab meeting.

Programming assignments (36%) Programming assignments must be completed individually. These will generally involve writing significant Java programs and may have "staged" due dates to ensure students progress throughout the assignment rather than waiting until the last minute to complete the task. Some of these assignments may involve writing analyses of the code and its behavior. Note: the last assignment does not involve writing any code; the only deliverable will be such an analysis.

Final Exam (15%)

The final exam will be cumulative. If you cannot take an exam at the scheduled time because of illness or other problems, you must contact me through Brightspace beforehand to arrange to take the exam at a different time. Failure to make prior arrangements for a missed exam will result in a grade of 0 for the exam.

In summary, grades are based on the following weighted items:

Item	% Weight
Attendance/Participation	10%
Laboratory Assignments	24%
Programming Assignments	36%
Quizzes	15%
Final exam	15%
Total	100%

Grade Ranges

Letter grades are assigned based on the final percent using the interval values:

Grade	% bound
A	93 - 100
A-	90 - 92.9
B+	87 - 89.9
B	83 - 86.9
B-	80 - 82.9
C+	77 - 79.9
C	73 - 76.9
C-	70 - 72.9
D	60 - 69.9
F	<60

The grade, “I” (incomplete), can be given ONLY when a student, who is doing otherwise acceptable work (passing grade), is unable to complete a part of work (e.g., the final exam) because of documented illness or other conditions beyond the student’s control. In the latter case, the student must discuss with the instructor and complete an application form from the department before the part of work is due or as soon as the circumstances are known.

Student Success Tips

- attend all class meetings
- read the material before coming to class
- complete assignments by the due dates specified
- create a study and/or assignment schedule to stay on track
- read announcements
- communicate regularly with your instructor and peers
- read and respond to course email messages as needed
- access USM Online Student Resources

Support Services

- **Request disability accommodations** | (207) 780-4706 | dsc-usm@maine.edu
- **Report interpersonal violence** | (207) 780-5767 | usm.titleix@maine.edu
- **Report on-campus emergencies and safety concerns** | (207) 780-5211 or your local police agency.
- **Get academic help** | mycampus.maine.edu/group/usm/learning-commons
- **Get technology help** | usm.maine.edu/computing/helpdesk
- **Meet with an academic advisor** | usm.maine.edu/advising

For USM’s most complete and current information on services available to students, as well as academic policies, use the QR Code to go to the Student Services and Policies Hub webpage.



Figure 1: <https://mycampus.maine.edu/group/usm/student-services-and-policies-hub>

Disclaimer

The syllabus is a statement of intent and serves as an implicit agreement between the instructor and the student. Every effort will be made to avoid changing the course schedule but the possibility exists that unforeseen events will make syllabus changes necessary. Remember to check Brightspace site often.
