CompST 751: Data Structures and Algorithms

John Boyland

Time and Location: TR 9:00-9:50am, EMS E190

EMS 925, 229-6986, boyland@uwm.edu
Office Hours: R 10:00-12:00, F 8:00-9:00

This course serves as introduction to data structure and algorithms via programming in an object-oriented language. It serves as a gateway into graduate study in Computer Science.

1 Description

Programming in a structured, high-level, object-oriented language. Implementation of data structures and algorithms and their application

2 Desired Outcomes

Students will be competent in object-oriented programming in Java, more specifically, they will be

• able to use and implement standard data structures (dynamic arrays, linked lists, binary search trees and hash tables), standard abstract data types (lists, tables, graphs, stacks, queues) and simple algorithms (bubble sort, binary search, depth-first search, breadth-first search);
• able to use inheritance and overriding to implement designs using dynamic-dispatch for polymorphism;
• able to use and implement generic containers and iterators following the design of the “collections” hierarchy;

3 Requirements

This course assumes that you are familiar with and have mastered the material from CompST 750 or CompSci 251. In particular, this class assumes students are

1. able to write, compile and execute Java programs;
2. able to use control structures, including methods, recursion and exceptions;
3. able to use standard datatypes including arrays, strings and array lists;
4. able to define classes with information hiding (abstract data types ADTs);
5. familiar with different kinds of testing.
4 Texts

The required textbook for the course is


Readings from the textbook are shown on the schedule. Additional readings (“lecture notes”) will be provided on the course website.

5 Grading

The grade for the course will be computed from the following parts:

40% Exams
There will be a mid-term examination and a final examination, each worth 20% of the grade. Exams will involve short answers on concepts, and programming done on paper. The final exam will be held at the designated time during finals week.

39% Programming Assignments
There will be a weekly programming assignment, except during midterm week. Each is worth 3% of the grade. The top thirteen scores will be used to compute the grade. Assignment due dates are firm—late homework will not be accepted.

11% Laboratory Assignments
Every Wednesday, we will post lab assignments that can be done on your computer; there will be short programming tasks, and questions to reflect on what is going on. If you wish, you may request attendance at an in-person lab where a TA will be present, assuming space is available. Unlike the undergraduate students, you will not be graded on the lab assignment. Instead, please find a recent paper in an ACM-sponsored journal or conference that covers a topic connected to the lab (“recent” means in the 2000s). Each week, you need to send the instructor the URL of the paper you found (which must be different every week). Please also write a paragraph explaining the connection.

10% Quizzes
There will at least twelve quizzes on the required reading. Some of the quizzes will be in class (not attending yields a zero) and others on the course website. The top ten quiz scores will be used for your grade.

(Other that the occasional in-class quiz, attendance is not directly assessed.) Exceptions to these rules can be made only in extraordinary circumstances. Advance notice of a need for an exception should be given whenever possible.

5.1 Criteria

Programming assignments will ask students to develop programs that implement tasks according to a provided specification. The programs may be required to use certain language features and libraries. Programs will be graded for functional correctness, suitability, style, clarity and practicality. Although we will provide solutions to some assignments, there are usually a wide variety of correct answers to any particular assignment.

5.2 Academic Honesty

The university has a responsibility to promote academic honesty and integrity and to develop procedures to deal effectively with instances of academic dishonesty. Students are responsible for the honest completion
and representation of their work, for the appropriate citation of sources, and for respect of others’ academic endeavors. A more detailed description of Student Academic Disciplinary Procedures may be found at http://uwm.edu/academicaffairs/wp-content/uploads/sites/32/2015/02/uws14facdoc1686.pdf

5.3 Course Grades
At the end of the course, the numeric grade will be converted into a letter grade according to the following scale:

<table>
<thead>
<tr>
<th>Letter Grade</th>
<th>A</th>
<th>A-</th>
<th>B+</th>
<th>B</th>
<th>B-</th>
<th>C+</th>
<th>C</th>
<th>C-</th>
<th>D+</th>
<th>D</th>
<th>D-</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Score</td>
<td>92</td>
<td>90</td>
<td>87</td>
<td>82</td>
<td>80</td>
<td>77</td>
<td>72</td>
<td>70</td>
<td>67</td>
<td>62</td>
<td>60</td>
<td>0</td>
</tr>
</tbody>
</table>

There is no curve, but the instructors reserve the right to increase a grade if they believe it would not reflect the student’s mastery of the material. A grade can only be decreased for academic dishonesty (following UWM academic misconduct disciplinary procedures).

6 Workload
In order to succeed in this course, we expect students will need to attend lecture (45 hours), do the programming assignments which will take about 8 hours each (100 hours), do the reading at about 12 pages an hour (40 hours) for a total of 185 hours of work.

7 Schedule
The anticipated weekly schedule (with required readings) is

1. Abstract Data Types (Chapter 2)
2. Dynamic Arrays (Chapter 3)
3. Generics, Writing Iterators (Chapter 5)
4. Linked Lists (Chapter 4)
5. Linked-List Variations (lecture notes)
6. Stacks& Queues (Chapter 6 and 7)
7. Mid-Term Examination (both days)
8. Binary Search Trees (Chapter 9)
9. Maps (Chapter 5.7, Oracle’s Map tutorial)
10. Hashing (Chapter 11)
11. Sorting (Chapter 12)
12. Graphs (Chapter 14)
13. XML (lecture notes)

8 Notes
If you will be needing any accommodation in this course for any reason, please contact the instructor. Please also be aware of the standard University policies: