CSCI 152 Interdisciplinary Computer Science II
Spring 2021

Class meets:  Monday, Wednesday  11:00 a.m. – 12:00 p.m.
              Zoom Lab:  Thursday 11 – 12
              In-person Lab:  Friday 11 – 12 (please register for and attend either lab section)

Professor:   Dr. Melissa Holmes
email:       melissa.holmes@umontana.edu
office:      https://umontana.zoom.us/my/melissaholmes
text messages:  406.565.6079
office hours: Updated as needed, and posted on Moodle

Course Description:
This class will build on knowledge constructed in CSCI 135 and CSCI 151. The Java programming language will be
introduced along with some object-oriented programming concepts and introductory data structures concepts.

Grading:
Short programming assignments, quizzes and problems  70%
Exams (2)                                          20%
Final Project                                     10%

Textbook:
None required. A variety of online resources will be used.

Accommodations:
Students who need any type of accommodation should work with Student Disability Services and provide
appropriate documentation as soon as possible.

Academic Dishonesty:
You are encouraged to work in teams and use many resources including books and the Internet. However, each
student must turn in his/her own work, and each student is responsible for understanding anything that is turned in.
Refer to the Student Conduct Code for more information regarding plagiarism and cheating.

Student Learning Outcomes: Upon the successful completion of this class, students will be able to:
1.  read a problem specification and define functional requirements for the problem;
2.  design a program to elegantly implement requirements;
3.  write Java programs of intermediate complexity;
4.  articulate introductory Data Structures & Algorithms concepts;
5.  use tools such as the JVM, command line compilation, and a modern IDE.
## Course Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong> Jan. 11-15</td>
<td>Syllabus, IDE and JDK, intro to Java, Hello World, Input/Output, short problems</td>
</tr>
<tr>
<td><strong>2</strong> Jan. 18-22</td>
<td>No class Monday Variables, arithmetic, Payroll program Decision statements, fortune picker program</td>
</tr>
<tr>
<td><strong>3</strong> Jan. 25-29</td>
<td>Loops / loop problems Average of die tosses (random)</td>
</tr>
<tr>
<td><strong>4</strong> Feb. 1-5</td>
<td>Methods - types, parameters Arrays – declaration, traversals</td>
</tr>
<tr>
<td><strong>5</strong> Feb. 8-12</td>
<td>Methods and Arrays, pass by value/reference, 2D arrays, exam review ZombieSim (no efficiency improvements)</td>
</tr>
<tr>
<td><strong>6</strong> Feb. 15-19</td>
<td>No class Monday Midterm Exam, more ZombieSim</td>
</tr>
<tr>
<td><strong>7</strong> Feb. 22-26</td>
<td>Intro OOP / ADT Intro UML, simple classes and inheritance</td>
</tr>
<tr>
<td><strong>8</strong> Mar. 1-5</td>
<td>Arrays of objects Object-Oriented ZombieSim</td>
</tr>
<tr>
<td><strong>9</strong> Mar. 8-12</td>
<td>Lists in Java (ArrayList, LinkedList, Stack, Queue) Implement ArrayList ADT</td>
</tr>
<tr>
<td><strong>10</strong> Mar. 15-19</td>
<td>Intro to Data Structures topics Big-O, Linear Search, Binary Search, File Input</td>
</tr>
<tr>
<td><strong>11</strong> Mar. 22-26</td>
<td>Implement a linked list (not using Java’s linked list class) that maintains a sorted list of integers – insert, remove, find</td>
</tr>
<tr>
<td><strong>12</strong> Mar. 29 – Apr. 2</td>
<td>No class (lab) Friday Circular Linked Lists, Doubly-Linked Lists, game simulation using LL</td>
</tr>
<tr>
<td><strong>13</strong> Apr. 5-9</td>
<td>Intro to sorting algorithms: Bubble sort, Insertion sort, Selection sort Sorting quiz / problems</td>
</tr>
<tr>
<td><strong>14</strong> Apr. 12 - 16</td>
<td>Class meetings TBA Final exam problem: Improve on ZombieSim or ArrayList; or complete an original programming problem. Final exam questions will be posted on Moodle.</td>
</tr>
<tr>
<td><strong>15</strong> Apr. 19 – 23</td>
<td>Class meetings TBA Final exam questions due Friday at 5:00 p.m.</td>
</tr>
<tr>
<td><strong>16</strong> Apr. 26 – 30</td>
<td>Final Exams Week Appointments to present your program and efficiency improvements</td>
</tr>
</tbody>
</table>
CSCI 152 Coding Standard

Comments

There should be 3-5 lines of comments at the top of each file. This heading should include your name, the class and semester, the lab # and title and the date. If it is part of a package you can include more information as needed.

Example:

/*     Melissa Holmes
       CSCI 152 Fall 2020
       Lab #1 Fortune Picker Lab     January 31, 2021
*/

Short comments throughout the code should describe the tasks of the program, i.e. “get n1 from the user”, purpose of variables, and the like. However, do not over-comment if things are obvious – comments are supposed to make your code easily readable by others.

Whitespace

A line of whitespace should exist between the tasks of the program.

Indenting should be used to indicate nesting in blocks of code. Spaces, not tabs, should be used to indent. Modern IDE’s have a setting to save whitespace as spaces.

Braces

When curly braces are used to indicate blocks of code, the first brace should appear on the same line as the code, and the closing brace should appear on its own line, i.e.:

    if (condition)   {
        //lines of code
    }  //closing brace on its own line

If appropriate for clarity, closing braces should be labeled with comments.

Identifier Names

Use identifier names that are meaningful, i.e. “firstSelection” may have more meaning than “a” or “n1”

Camel case should be used for variable and method names, i.e. theFirstVariable, myMethod

Camel case with the first letter capitalized should be used for class names, i.e. HelloWorld