CSCI136 Fundamentals of Computer Science II
Spring 2020

Class meets: Monday, Wednesday 2:00 – 2:50 p.m.
Labs are held Thursday 11:00 – 11:50 p.m. and Friday 2:00 – 2:50 p.m.

Professor: Dr. Melissa Holmes
email: melissa.holmes@umontana.edu
office: Social Sciences 411
text messages: 406.565.6079
office hours: Monday 12:00 – 2:00 p.m.
Thursday 1:00 – 4:00 p.m.
Friday 12:00 – 2:00 p.m.
and by appointment (or see if my door is open)

Course Description:
This class will build on knowledge constructed in CSCI135. The Java programming language will be introduced along with some object-oriented programming concepts. The class helps prepare you for the Data Structures & Algorithms course by introducing some elementary data structures, and introduces GUI programming with the JavaFX libraries.

Grading:
Short programming assignments, quizzes and problems 60%
Exams (2) 20%
Large programming project 20%

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 a UML diagram (API) that matches an implemented class hierarchy or a program specification, and implement a class that matches a UML diagram;
4. explain and illustrate the terms inheritance, polymorphism, encapsulation, and abstraction, and write programs making correct use of all these concepts;
5. use exception handling appropriately; and
6. use tools such as the JVM, command line compilation, and a modern IDE.
# Course Schedule (Subject to Change)

<table>
<thead>
<tr>
<th>Week</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jan 13-17</td>
</tr>
</tbody>
</table>
| 2     | Jan 20-24  | No classes Monday  
Assignment, decision, repetition constructs in Java |
| 3     | Jan 27-31  | Methods and method calls |
| 4     | Feb 3-7    | Arrays – declaring, traversals, exception handling |
| 5     | Feb 10-14  | Arrays, passing arrays as parameters |
| 6     | Feb 17-21  | No classes Monday  
Introduction to classes and objects |
| 7     | Feb 24-28  | Exam 1 Wednesday  
Classes and objects continued |
| 8     | Mar 2-6    | Processing arrays of objects |
| 9     | Mar 9-13   | Processing arrays of objects – disease spread simulation |
| 10    | Mar 16-20  | Spring Break – no classes |
| 11    | Mar 23-27  | Lists and collections of objects in Java  
Linked list implementation |
| 12    | Mar 30 – Apr 3 | Linked list continued  
Stacks and Queues |
| 13    | Apr 6-10   | Introduction to JavaFX  
Final Project Specification |
| 14    | Apr 13-17  | JavaFX, Final Project |
| 15    | Apr 20-24  | JavaFX, Final Project |
| 16    | Apr 27-May 1 | Final Project, Final Exam distributed |
| 17    | May 4-8    | Final Exams Week  
Final Exam Due Wednesday, May 20 at 5:20 p.m., Final Project Demo (schedule) |
CSCI136 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
   CSCI136 Fall 2019
   Lab #1 Fortune Picker Lab September 4, 2019
*/

Short comments throughout the code should describe the tasks of the program, i.e. “get n1 from the user”

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