Overview

The course is an introduction to the art and science of computer programming and related computer science principles. Through programming students will develop fundamental skills such as abstract reasoning and problem solving. Students will master programming techniques using the Java programming language, and will develop good program design methodology resulting in correct, robust, and maintainable programs.

Prerequisites

No previous background in programming is required, only dedication and hard work.

Learning Objectives

The objective of the course is to:

- Learn programming terminology and obtain a solid grasp of the basic mechanics of programming.
- Understand code and be able to trace the execution of code fragments.
- Analyze and explain the behavior of programs involving the fundamental programming constructs covered during lecture.
- Design, implement, test, and debug programs that use each of the fundamental programming constructs covered during lecture.
- Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
- Describe the mechanics of parameter passing.
- Understand the basics of object-oriented concepts.

Textbook

  
  - The textbook is optional, no assignments or required readings will be given directly from the textbook, so you may choose not to purchase it if you like. However, the book makes a useful supplement to the lecture presentations. It contains practice problems and online videos you can use to study for your exams.
- Lecture notes and sample programs will be posted on Latte.
Software

The recommended software for the course is the Java Development Kit (JDK) version 8 and a plain text editor.

Grading

The final grades for the course will be determined using the following weights:

- Homework: 40%
- Quizzes and Exam: 50%
- Participation: 10%

- Homework consists of 7 programming assignments done individually. Programs will be graded on “external correctness” (behavior) and “internal correctness” (style and design). Late homework will not receive credit.
- Make-up exam and quizzes will not be given except in case of a serious emergency.

Academic Honesty

As stated in the Rights and Responsibilities handbook, "Every member of the University community is expected to maintain the highest standards of academic honesty. A student shall not receive credit for work that is not the product of the student's own effort."

Programming assignments must be completed individually (unless specified otherwise by the instructor); all code you submit must be your own work. You may discuss general ideas of how to approach an assignment, but never specific details about the code to write. Any help you receive from or provide to classmates should be limited and should never involve details of how to code a solution.

As a student of this course you are agreeing to the following rules:

- You may not work as a partner with another student on an assignment.
- You may not get code from online sources.
- You may not show another student your solution to an assignment, nor look at his/her solution, for any reason.
- You may not have another person "walk you through" an assignment, describe in detail how to solve it, or sit with you as you write it. You also may not provide such help to another student. This includes current or former students, tutors, friends, TAs, web site forums, or anyone else.
- You may not post your homework solution code online or ask others for online help. This includes public message boards, forums, file sharing sites and services, or any other online system.

Under our policy, a student who gives inappropriate help is equally guilty with one who receives it. Instead of providing such help to someone who does not understand an assignment, point him or her to other class resources such as lecture examples, the textbook, or emailing a TA or instructor. You must not share your solution and ideas with others. You must also ensure that your work is not copied by others, such as making sure to log out of shared computers, not leaving printouts of your code in public places, and not emailing your code to other students or posting it on the web. We enforce this policy by running similarity detection software over all submitted student programs.
### Course Outline

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Thu</th>
<th>Tentative Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>June</td>
<td>6</td>
<td>7</td>
<td>Lecture 1</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>14</td>
<td>Lecture 4</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>21</td>
<td>Lecture 7</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>28</td>
<td>Lecture 10</td>
</tr>
<tr>
<td>July</td>
<td>4</td>
<td>5</td>
<td>Lecture 13</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>28</td>
<td>Lecture 11</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td>Lecture 12</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>Lecture 13</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>Lecture 14</td>
</tr>
</tbody>
</table>

- Intro to Java Programming
  - Basic Computing Concepts
  - Program Errors
  - Procedural Decomposition
- Primitive Data Types
- The for Loop
- Parameters and Objects
- Conditional Execution
- Text Processing
- The while Loop
- Random Numbers
- File Processing
- Arrays
- Array-Traversal Algorithms
- Sorting and Searching
  - Bubble-sort, Selection Sort
  - Linear Search, Binary Search
- Recursion
- Classes and Objects
- Constructors
- Encapsulation
- Inheritance Basics
- Inheritance
- Polymorphism

**Note:** topics are tentative