ICS111 – Introduction to Computer Science

3 Credits
Distance Learning

INSTRUCTOR: Laura Sue
OFFICE: Hale Palanakila 119A
OFFICE HOURS: Online
TELEPHONE: 236-9253
EMAIL: laurasue@hawaii.edu
EFFECTIVE DATE: Summer 2018
CRN: 63045

Windward Community College Mission Statement

Windward Community College offers innovative programs in the arts and sciences and opportunities to gain knowledge and understanding of Hawai‘i and its unique heritage. With a special commitment to support the access and educational needs of Native Hawaiians, we provide O‘ahu’s Ko‘olau region and beyond with liberal arts, career and lifelong learning in a supportive and challenging environment — inspiring students to excellence.

Catalog Description

Intended for computer science majors and all others interested in a first course in programming. An overview of the fundamentals of computer science emphasizing problem solving, algorithm development, implementation, and debugging/testing using an object-oriented programming language.

Prerequisite: Credit for MATH 103 with a grade of “C” or better, placement into Math 135, or consent of instructor.

Student Learning Outcomes

1. Use an appropriate programming environment to design, code, compile, run and debug computer programs.
2. Demonstrate basic problem solving skills: analyzing problems, modeling a problem as a system of objects, creating algorithms, and implementing models and algorithms in an object-oriented computer language (classes, objects, methods with parameters, abstract classes, interfaces, inheritance and polymorphism).
3. Illustrate basic programming concepts such as program flow and syntax of a high-level general purpose language and basic security practices.
4. Demonstrate working with primitive data types, strings and arrays.
Course Tasks and Student Learning Outcomes Alignment

<table>
<thead>
<tr>
<th>Student Learning Outcomes</th>
<th>REVEL Exercises</th>
<th>Programming Assignments</th>
<th>Final Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an appropriate programming environment to design, code, compile, run and debug computer programs.</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Demonstrate basic problem solving skills: analyzing problems, modeling a problem as a system of objects, creating algorithms, and implementing models and algorithms in an object-oriented computer language (classes, objects, methods with parameters, abstract classes, interfaces, inheritance and polymorphism).</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Illustrate basic programming concepts such as program flow and syntax of a high-level general purpose language and basic security practices.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demonstrate working with primitive data types, strings, and arrays.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Assessment Tasks and Grading

**REVEL Exercises:** We will be using Pearson’s REVEL tool (https://revel.pearson.com) to access the textbook. REVEL includes videos, animations, quizzes, and programming exercises throughout the text that will create a more engaging and interactive learning experience for you. A nice feature for a programming course like this one is that you can practice writing code and get immediate feedback. This tool will allow you to write snippets of code to try out the concepts you are learning. It can provide you with hints if you don’t get the solution quite right. Each chapter will contain a new set of exercises to work on. The number of exercises will vary from chapter to chapter. As long as you complete at least 80% of all exercises correctly, you will get full credit for that REVEL assignment. Otherwise, your score in Laulima will be a percentage of the exercises you've completed successfully in REVEL. You will have an unlimited number of tries for each exercise, until the due date, to get the correct solution. The REVEL Exercises are worth 2 points each. The registration link for our course is https://console.pearson.com/enrollment/bwmudp.

**Programming Assignments:** There will also be a Programming Assignment for each chapter. In these assignments, you will write complete programs and submit them to Laulima. These assignments are designed to give you practice in writing full programs in a real programming environment. Programming Assignments are worth 3 points each.
Final Project: There are no exams in this class. Instead, you will complete a Final Project, which will encompass all concepts covered in the course. The Final Project will be due on the last day of the session: Friday, August 10, 2018 at 11:55 pm.

Late work: Because this is a summer class, there is no late period for any assignments. If you think you might not be able to submit any assignment on time, you must contact the instructor (as soon as possible, but at least 24 hours) before the due date to make arrangements to complete your assignment(s). If you do not contact the instructor before the due date, you will not be allowed to make up the work.

Assignment Breakdown:

<table>
<thead>
<tr>
<th>Assignments</th>
<th>Points</th>
<th>Percentage of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVEL Exercises</td>
<td>26</td>
<td>32%</td>
</tr>
<tr>
<td>Programming Assignments</td>
<td>39</td>
<td>48%</td>
</tr>
<tr>
<td>Final Project</td>
<td>16</td>
<td>20%</td>
</tr>
<tr>
<td>GRAND TOTAL</td>
<td>81</td>
<td>100%</td>
</tr>
</tbody>
</table>

Learning Resources

- **Laulima:** [https://laulima.hawaii.edu](https://laulima.hawaii.edu)
- **REVEL:** [https://revel.pearson.com](https://revel.pearson.com)
- **Software:**
  - Eclipse: [https://www.eclipse.org/](https://www.eclipse.org/)

Additional Information

**Business-like behavior:** ICS courses at Windward Community College are part of the Business department. In order to fulfill the objectives of the Business department, students are expected to present business-like behavior. Business-like behavior includes:

**Time-management:** Since this is a distance learning class, it will be up to you to schedule enough time to complete the lessons each week. Don’t wait until the last minute to complete assignments. This is true in almost any class, but especially when it comes to writing programs, you should give yourself plenty of time to figure out how to solve the problem.
**Turn in assignments on time:** Start assignments well before the due date. If situations arise which prevent assignments from being completed on time, notify the instructor right away.

**Ask for assistance:** In a business, if you were uncertain about what to do, you would ask your boss for direction. In this class, ask the instructor for assistance.

**Email:** Please use your UH email address for this course. Any information regarding the class will be sent to your UH email address, so check your email frequently. Email is also the preferred method of contacting the instructor.

**Academic Dishonesty:** Academic dishonesty includes, but is not limited to, file sharing (giving or receiving of files between students), more than one student working on the same file, and copying work in full or in part from another student or other sources such as the Internet. Any student caught cheating will automatically receive a 0 for the assignment. In addition, a report of the incidence will be filed, which may result in the student being expelled from the school. For more information, please see the college catalog for the school’s policy on academic dishonesty.

**MySuccess**

At Windward Community College we want every student to be successful. MySuccess is a system-wide effort that seeks to support students early in the semester when they first begin experiencing difficulty in class. If I feel that you’re having difficulty in my class within the first few weeks of the semester (e.g. missing assignments, or low scores) and working together to address your challenges shows that you would really benefit from being connected to resources outside of the classroom, I may refer you to your assigned counselor. Once referred, MySuccess will:

- Call you and send an email to your hawaii.edu account to let you know about my referral; and
- Have a Counselor follow up with you by phone or by email to find out what kinds of help you might need and connect you with the necessary resources to help you devise a strategy for success.

I will not refer you without telling you. However, if I do refer you, know that I am doing so in an effort to connect you with all of the help you may need to do well this semester as your success is important to me.

**Disabilities Accommodation Statement**

If you have a physical, sensory, health, cognitive, or mental health disability that could limit your ability to fully participate in this class, you are encouraged to contact the Disability Specialist Counselor to discuss reasonable accommodations that will help you succeed in this class. Ann Lemke can be reached at 235-7448, lemke@hawaii.edu, or you may stop by Hale ʻAkoakoa 213 for more information.
# Tentative Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
</table>
| Week 1    | Introduction to ICS 111  
Chapter 1 – Introduction to Computers, Programs, and Java  
Chapter 2 – Elementary Programming  
Chapter 3 – Selections |
| 7/2/2018  |                                                                      |
| Week 2    | Chapter 4 – Mathematical Functions, Characters, and Strings  
Chapter 5 – Loops  
Chapter 6 – Methods |
| 7/9/2018  | **Week 1 & 2 Assignments due Sunday 7/15/2018**                      |
| Week 3    | Chapter 7 – Single-Dimensional Arrays  
Chapter 8 – Multidimensional Arrays  
Chapter 9 – Objects and Classes |
| 7/16/2018 |                                                                      |
| Week 4    | Chapter 10 – Thinking in Objects  
Chapter 11 – Inheritance and Polymorphism |
| 7/23/2018 | **Week 3 & 4 Assignments due Sunday 7/29/2018**                      |
| Week 5    | Chapter 12 – Exception Handling and Text I/O  
Chapter 13 – Abstract Classes and Interfaces |
| 7/30/2018 |                                                                      |
| Week 6    | Final Project                                                             |
| 8/6/2018  | **Week 5 & 6 Assignments due Friday 8/10/2018**                       |

Please note that the schedule may change as necessary.
Course Content

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Use an appropriate programming environment to design, code, compile, run and debug computer programs.</strong>&lt;br&gt;a. Programming-tools.&lt;br&gt;1) Integrated Development Environment (IDE) or a text editor and command line-based compilation and execution.&lt;br&gt;b. Coding a solution.&lt;br&gt;1) Self-documenting programs.&lt;br&gt;2) Good formatting.&lt;br&gt;c. Compile and run programs.&lt;br&gt;d. Debug programs.</td>
<td><strong>1. Use an appropriate programming environment to design, code, compile, run and debug computer programs.</strong>&lt;br&gt;a. Use programming tools to model a problem and design algorithms that express its solution.&lt;br&gt;b. Formulate models and algorithms in the syntax of an object-oriented programming language using either an Integrated Development Environment (IDE) or a text editor.&lt;br&gt;c. Utilize either an IDE or a command prompt to compile and run programs.&lt;br&gt;d. Test and debug programs to produce code that runs and generates the correct results.</td>
</tr>
</tbody>
</table>

| **2. Demonstrate basic problem solving skills: analyzing problems, modeling a problem as a system of objects, creating algorithms, and implementing models and algorithms in an object-oriented computer language (classes, objects, methods with parameters, abstract classes, interfaces, inheritance and polymorphism).**<br>a. Analysis of a problem by identifying objects and classifying them.<br>b. Design a solution to the problem by defining the messages objects send each other, the parameters the messages carry and the inheritance among object classes.<br>c. Classes, objects, and methods.<br>1) Classes objects, and methods described.<br>a) Classes.<br>b) Objects.<br>c) Method declarations and method calls<br>d) Overloaded methods.<br>2) Incorporate parameter passing.<br>a) Formal and actual parameters.<br>b) Returning values from methods<br>c) Parameter passing by value and by reference.<br>3) Write simple classes and objects.<br>a) Classes.<br>b) Objects.<br>c) Method declaration/implementation and method calls.<br>d) Constructors.<br>e) Encapsulation through visibility modifiers (public, private)<br>f) Class and instance methods and fields. | **2. Demonstrate basic problem solving skills: analyzing problems, modeling a problem as a system of objects, creating algorithms, and implementing models and algorithms in an object-oriented computer language (classes, objects, methods with parameters, abstract classes, interfaces, inheritance and polymorphism).**<br>a. Classes, objects, and methods<br>1) Use API classes, objects, and methods, citing examples.<br>2) Write simple classes and create objects that interact between multiple classes.<br>3) Understand parameter passing and methods returning values<br>4) Inheritance and Polymorphism<br> a) Model a problem as a hierarchy of classes<br>b) Differentiate between overloading and overriding.<br>5) Define Interfaces and implement them with classes<br>b. Apply problem-solving techniques such as stepwise refinement and object-oriented analysis<br>c. Incorporate the concept of software life cycle into program development.<br>d. Determine and design an algorithm to solve a specific problem.<br>e. Evaluate algorithm performance. |
3. **Illustrate basic programming concepts such as program flow and syntax of a high-level general purpose language and basic security practices.**
   a. Sequence.
   b. Selection.
   c. Repetition.

4. **Demonstrate working with primitive data types, strings and arrays.**
   a. Primitives Types
      1. Numeric, character and boolean types.
      2. Numeric accuracy.
      3. Memory requirements.
      4. Declaration.
      5. Initialization.
   b. Integer Arithmetic
      1. Addition and subtraction, increment and decrement
      2. Multiplication, division, and modulo.
      3. Truncation.
   c. Casting
      1. Type assignment.
      2. Implicit and explicit casting.
   d. Strings
      1. Constants
      2. Concatenation.
   e. Arrays
      1. Declaration
      2. Access to array vs. access to an element
      3. Multidimensional arrays