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

This is an introductory course for students intending to major in computer science and requiring a computer programming course. Emphasis will be on problem solving, algorithm/pseudocode development, structured programming, computer language coding, implementation and debugging/testing. Students will develop application programs in an IBM microcomputer/DOS/Windows operating system environment. Students will be taught to develop appropriate programs using accepted standards and methodologies. Actual programming is a part of this course.

Prerequisite: Credit for MATH 103 or higher; 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.
4. Identify relationships between computer systems programming and programming languages.
5. Demonstrate working with primitive data types, strings and arrays.
<table>
<thead>
<tr>
<th>Student Learning Outcomes</th>
<th>Assignments</th>
<th>Exams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an appropriate programming environment to design, code, compile, run and debug computer programs</td>
<td>All programming assignments</td>
<td>Midterm and final exams</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>All programming assignments</td>
<td>Midterm and final exams</td>
</tr>
<tr>
<td>Illustrate basic programming concepts such as program flow and syntax of a high-level general purpose language.</td>
<td>All programming assignments</td>
<td>Midterm and final exams</td>
</tr>
<tr>
<td>Identify relationships between computer systems programming and programming languages</td>
<td>All programming assignments</td>
<td>Midterm and final exams</td>
</tr>
<tr>
<td>Demonstrate working with primitive data types, strings and arrays</td>
<td>All programming assignments</td>
<td>Midterm and final exams</td>
</tr>
</tbody>
</table>
Assessment Tasks and Grading

<table>
<thead>
<tr>
<th>Assignments</th>
<th>Points</th>
<th>Percentage of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classwork</td>
<td>50</td>
<td>10%</td>
</tr>
<tr>
<td>Homework</td>
<td>300</td>
<td>60%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>75</td>
<td>15%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>75</td>
<td>15%</td>
</tr>
<tr>
<td><strong>GRAND TOTAL</strong></td>
<td><strong>500</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

Grades for the course will be as follows:

- **A** 90-100% of possible points
- **B** 80-89% of possible points
- **C** 70-79% of possible points
- **D** 60-69% of possible points
- **F** 0-59% of possible points

Learning Resources

- **Laulima**: [https://laulima.hawaii.edu/portal](https://laulima.hawaii.edu/portal)
- **Software**:

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:

- **Attendance**: Attend class regularly. This includes arriving on time and remaining until the end of class. Students who attend class regularly are more likely to earn higher grades.

- **Online Discussions**: Be courteous in online discussion areas.

- **Turn in assignments on time**: Start assignments 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.
**Absences:** Students who are ill or have other reasons for missing class should email the instructor for an excused absence. The student is responsible for the material covered in class and any in-class work missed. Any assignments due at the beginning of class should be turned in online or at the start of the next class.

**Late work:** Assignments may be submitted up to one week late for an automatic penalty of 10%.

**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 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.

**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 ‘Akoakoa 213 for more information.

**Course Content**

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Skills</th>
</tr>
</thead>
</table>
| **1. Use an appropriate programming environment to design, code, compile, run and debug computer programs.**  
a. Programming-tools.  
   1. Integrated Development Environment (IDE) or a text editor and command line-based compilation and execution.  
b. Coding a solution.  
   1. Self-documenting programs.  
   2. Good formatting.  
c. Compile and run programs.  
   a. Debug programs.  
| **1. Use an appropriate programming environment to design, code, compile, run and debug computer programs.**  
a. Use programming tools to model a problem and design algorithms that express its solution.  
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.  
c. Utilize either an IDE or a command prompt to compile and run programs.  
d. Test and debug programs to produce code that runs and generates the correct results.  
| **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, **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,
objects, methods with parameters, abstract classes, interfaces, inheritance and polymorphism).

a. Analysis of a problem by identifying objects and classifying them.
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.
c. Classes, objects, and methods.
   1) Classes objects, and methods described.
      a) Classes.
      b) Objects.
      c) Method declarations and method calls
      d) Overloaded methods.
   2) Incorporate parameter passing.
      a) Formal and actual parameters.
      b) Returning values from methods
      c) Parameter passing by value and by reference.
   3) Write simple classes and objects.
      a) Classes.
      b) Objects.
      c) Method declaration/implementation and method calls.
      d) Constructors.
      e) Encapsulation through visibility modifiers (public, private)
      f) Class and instance methods and fields (static)
   4) Inheritance and Polymorphism
      a) Extending classes, subclasses
      b) Overriding methods
      c) Polymorphism
   5) Interfaces
      a) Interfaces as types
      b) Implementing by classes
   6) Program Development
      a) Algorithm design and representation using pseudocode, flowcharts, etc.
      b) Evaluate algorithm efficiency.
      c) Stepwise refinement.
      d) Program lifecycle.

b. Apply problem-solving techniques such as stepwise refinement and object-oriented analysis

c. Incorporate the concept of software life cycle into program development.
d. Determine and design an algorithm to solve a specific problem.
e. Evaluate algorithm performance.

3. Illustrate basic programming concepts such as program flow and syntax of a high-level general purpose language.
a. Sequence.
b. Selection.

3. Illustrate basic programming concepts such as program flow and syntax of a high-level general purpose language.
a. Describe sequential, branching, and repetitive concepts.
4. **Identify relationships between computer systems, programming and programming languages.**
   a. Computer organization and architecture (memory, arithmetic-logic unit, control unit).
   b. Binary representation of data (range of data type, precision and round-off, image representation).
   c. Operating system concepts.
   d. Programming language assembler/compiler.

4. **Identify relationships between computer systems, programming and programming languages.**
   a. Examine the hardware (binary numbers, character encoding, Boolean logic) and basic computer system architecture concepts.
   b. Examine system software and virtual machine concepts.
   c. Describe the concept of program compilation and translation to machine code.

5. **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

5. **Demonstrate working with primitive data types, strings and arrays.**
   a. Primitive types
      1) Utilize and understand primitive types, their accuracy, memory requirements
      2) Declarations and initialization of primitive types.
      3) Demonstrate integral arithmetic including mod.
      4) Explain casting and differentiate between implicit and explicit casting.
   b. Strings
   c. Arrays