

ICS 211
Introduction to
Computer Science II

```
format.xml] { redirect_to(ope  
{ head :ok }  
else  
format.html] { render :acti
```

ICS 211 - Introduction to Computer Science II

Instructor Information

David Maxson

David.Maxson@hawaii.edu

Office hours: Online only

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

Reinforce and strengthen problem solving skills using more advanced features of programming languages and algorithms such as recursion, pointers, and memory management. Emphasize the use of data structures such as arrays, lists, stacks and queues.

Student Learning Outcomes

The Student Learning Outcomes for this course are:

- Recognize the use of arrays, lists, stacks, queues, and other data structures.
- Select the appropriate searching and sorting algorithm based on the algorithm's behavior.
- Develop recursive algorithms and programs.
- Select appropriate data structure for a given application.
- Use advanced object-oriented programming techniques (polymorphism, inheritance, and encapsulation) and standard libraries.
- Produce robust programs using exception handling and extensive program testing.
- Create simple graphical user interface (GUI) program.

Student Learning Outcomes Alignment

Student Learning Outcome	Lessons and Assessments
Recognize the use of arrays, lists, stacks, queues, and other data structures.	Lessons 4, 5, 6 Assignments 4, 5, 6 14
Select the appropriate searching and sorting algorithm based on the algorithm's behavior.	Lessons 4, 5, 6 Assignments 4, 5, 6, 14
Develop recursive algorithms and programs.	Lesson 3 Assignments 3, 6
Select appropriate data structure for a given application.	Lessons 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14 Assignments 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14
Use advanced object-oriented programming techniques (polymorphism, inheritance, and encapsulation) and standard libraries.	Lessons 9, 10, 11, 12, 13 Assignments 9, 10, 11, 12, 13
Produce robust programs using exception handling and extensive program testing.	Lessons 2, 8, 13 Assignments 2, 8, 13, 14
Create simple graphical user interface (GUI) program.	Lessons 9, 10, 11, 12, 13 Assignments 9, 10, 11, 12, 13

Course Content

Concepts	Skills
<p>1. Recognize the use of arrays, lists, stacks, queues, and other data structures.</p> <p>a. Select the appropriate searching or sorting algorithm based on the algorithm's behavior.</p> <ol style="list-style-type: none"> 1) Illustrate data types and characterize them. <ol style="list-style-type: none"> a) Data types. b) Characteristics. 2) Explain how data types are used in program control and corresponding potential pitfalls. <ol style="list-style-type: none"> a) Stacks. b) Queues. c) Use of pointers. 3) Demonstrate use of data structures by writing classes and incorporating built-in classes/libraries. 	<p>1. Recognize the use of arrays, lists, stacks, queues, and other data structures.</p> <ol style="list-style-type: none"> a. Illustrate data types and characterize them. b. Explain how data types are used in program control and potential pitfalls. c. Demonstrate use of data structures by writing classes and incorporating built-in classes/libraries. d. Use of pointers.

<ul style="list-style-type: none"> a) Data structures. b) Packages/libraries. c) Write classes 	
<p>2. Select the appropriate searching or sorting algorithm based on the algorithm's behavior.</p> <ul style="list-style-type: none"> a. Sorting algorithms (e.g.: selection sort, insertion sort, bubble sort, quick sort). b. Searching algorithms (sequential vs. binary). 	<p>2. Select the appropriate searching or sorting algorithm based on the algorithm's behavior</p> <ul style="list-style-type: none"> a. Analyze the efficiency of various sorting algorithms, including insertion sort, selection sort, bubble sort, merge sort, and quicksort. b. Analyze the efficiency of various searching algorithms including linear search and binary search. c. Utilize an appropriate sorting technique for a given data set. d. Utilize an appropriate searching technique for a given data set.
<p>3. Develop recursive algorithms and programs.</p> <ul style="list-style-type: none"> a. Recursion concept. b. Recursion implementation. c. Iterative vs. recursive efficiency.. 	<p>3. Develop recursive algorithms and programs.</p> <ul style="list-style-type: none"> a. Write recursive functions and algorithms. b. Compare efficiency of iterative vs. recursive solution.
<p>4. Select appropriate data structure for a given application.</p> <ul style="list-style-type: none"> a. Performance characteristics of array and linked implementations. <ul style="list-style-type: none"> 1) Performance evaluation. b. Problem characteristics to determine whether array, list, stack, or queue provides best representation. c. Problem analysis for data structure selection.. 	<p>4. Select appropriate data structure for a given application.</p> <ul style="list-style-type: none"> a. Choose between array and linked implementation. b. Evaluate problem characteristics to determine whether array, list, stack, or queue provides best representation..
<p>5. Use advanced object-oriented programming techniques such as inheritance and standard libraries.</p> <ul style="list-style-type: none"> a. Inheritance. b. Standard libraries. c. Library organization 	<p>5. Use advanced object-oriented programming techniques such as inheritance and standard libraries.</p> <ul style="list-style-type: none"> a. Write a derived class. b. Navigate the library hierarchy. c. Use appropriate library functions
<p>6. Produce robust programs using exception handling and extensive program testing.</p> <ul style="list-style-type: none"> a. Exception handling. b. Program testing techniques. c. Testing design strategies. d. Defensive programming.. 	<p>6. Produce robust programs using exception handling and extensive program testing</p>
<p>7. Create simple graphical user interface (GUI).</p> <ul style="list-style-type: none"> a. Good user interface design. b. Library GUI objects. c. Event-driven programming paradigm. 	<p>7. Create simple graphical user interface (GUI)</p> <ul style="list-style-type: none"> a. Demonstrate knowledge of available GUI objects. b. Design a simple GUI. c. Implement a GUI program with event-driven programming techniques.

Course Tasks

In this class, you must show mastery of each concept through a series of projects. Every assignment is worth 3 points except the final project, which is worth 9 points.

To get the full 3 points, your work must not contain any errors. If there are errors, then I will specify what it is and return it to you. Note that I will not tell you how to correct the error, only what it is. You should then correct the assignment and resubmit it.

The only deadline for all assignments is May 6, 2015. No work will be accepted after that date. This is a firm deadline.

Assignment Tasks and Grading

Your letter grade will be determined by the number of assignments you complete. There are 13 assignments and the final project for a total of 48 points:

- A: 43 - 48 points
- B: 38 - 42 points
- C: 33 - 37 points
- D: 28 - 32 points
- F: 0 - 27 points

Learning Resources

We will be using *Imagine! Java: Programming Concepts in Context* by Frank M. Carrano as our textbook.

You need to be comfortable with using the documentation for the Java and JavaFX API's. You will be referring to the API's to become familiar with the libraries available through the JDK's.

We will use Laulima for submitting and returning all assignments. All grades will be posted in Laulima and you will be able to track your progress by utilizing the grade book. You will be able to post and read questions and comments on the discussion boards. Use the private message tool in Laulima to contact the instructor.

We will be using the Java programming language to develop our programs. Go to the [Java Download Page](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html) (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>) to download the latest SE JDK. It is vital that you install it correctly, including setting the correct path environment.

Although it isn't necessary, I also recommend you use an Integrated Development Environment such as [jGrasp](http://jgrasp.org) (<http://jgrasp.org>). I do not recommend using NetBeans or Eclipse at this time. Both insert code in

your projects that could keep it from compiling from the command line.

As an alternative, you can use uhunix (type `ssh username@uhunix.hawaii.edu` at the command line. Replace username with your UH username) or an online development site such as [Cloud9 IDE \(https://c9.io\)](https://c9.io).

Other resources

Tutoring may be available from the TRIO office in the Library Learning Commons on the WCC campus.

Policies

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. Ann Lemke can be reached by phone at 235-7448, by email at lemke@hawaii.edu, or by stopping by her office in Hale 'Akoakoa 213 for more information.

Academic Dishonesty - Cheating and Plagiarism

You are responsible for the content and integrity of all work you submit. The guiding principle of academic integrity will be that all files, work, reports, and projects that you submit are your own work.

You will be guilty of cheating if you:

- Represent the work of others as your own (plagiarism).
- User or obtain unauthorized assistance in any academic work.
- Give unauthorized assistance to other students.
- Modify, without instructor approval, an examination, paper, record, or report for the purpose of obtaining additional credit.
- Misrepresent the content of submitted work.

Netiquette

Whenever you post something to the discussion board or other online forums, you are expected to follow proper netiquette. Be respectful at all times. Do not use obscene language or make disparaging comments, even if it is meant as a joke. Remember that others cannot see your facial expression nor hear your tone of voice, so they will not know you are trying to be witty. Do not use all caps. Using all caps is normally interpreted to be shouting.

Discussion Boards

Discussion boards are to be used for class work only. Do not post political or other comments or statements, nor solicit sales for any type of product. Your instructor will be monitoring all communication in Lualima and will take appropriate action when necessary

A Final Thought

All programming languages use the same basic concepts. By learning the concepts and writing your initial program in pseudocode, you should be able to use any programming language to code your software. It is vital that you understand these concepts. You will use them throughout your studies in Computer Science and as a programmer or Software Engineer afterward. The best way to learn them is to use them. Try designing and creating programs that you will find useful. Good luck!