CodeHS

Arizona Software and App Design II Syllabus
1 year for High School (160-170 contact hours)

Introduction

The Arizona Software and App Design II Course is intended to teach students the fundamentals of developing, implementing, and evaluating computer software and program applications. Students will be using the Java programming language as they develop efficient algorithms, data structures, error handling techniques, and version control best practices. Over the course of the school year, students will collaboratively and independently design, develop and implement programs using these foundational skills.

Course Overview and Goals

Prerequisites

The Arizona Software and App Design II course is the second course of the Arizona Software and App Design pathway. Students should first complete Arizona Software and App Design I before enrolling in this course.

Learning Environment

The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 100 hours of hands-on programming practice in total. Several units have free response questions that have students consider the applications of programming and incorporate examples from their own lives.

Programming Environment

Students write and run Java programs in the browser using the CodeHS editor.

More information: Browse the content of this course at https://codehs.com/course/20492

Course Breakdown

(Optional) Unit 1: Introduction to Programming in Java with Karel the Dog (3 weeks/15 hours)
Students learn the basics of java commands, control structures, and problem solving by solving puzzles with Karel. This is an optional unit 1 for students who would benefit from a Karel introduction to programming in Java. This unit can be found in the supplemental material of the course. If students do not need this unit, then starting with Networks and the Internet as unit 1 is recommended.

Topics Covered	• Commands • Defining vs. Calling Methods • Designing methods • Program entry points • Control flow • Looping • Conditionals • Classes • Commenting code • Preconditions and Postconditions

Unit 2: Networks and the Internet (2 weeks/10 hours)
This unit explores the structure and design of the internet, and how this design affects the reliability of network communication, the security of data, and personal privacy.

Topics Covered	• Structure of the internet • How network data is transmitted • Hardware involved in the transmission of data • How the internet has impacted everyday life

Unit 3: Primitive Types (2.5 weeks/10 hours)
This unit introduces students to the Java programming language and the use of classes, providing students with a firm foundation of concepts that will be leveraged and built upon in all future units. Students will focus on writing the main method and will start to call preexisting methods to produce output.

Topics Covered	• Why programming? Why Java? • Variables and Data Types • Expressions and Assignments Statements • Compound Assignment Operators • User Input • Casting and Ranges of Variables

Unit 4: Using Objects (3 weeks/15 hours)
This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer's purpose. This unit builds on students' ability to write expressions by introducing them to `Math` class methods to write

expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the `String` class are an important topic within this unit.

Topics Covered	• Objects: Instances of Classes • Creating and Storing Objects (Instantiation) • Calling a Void Method • Calling a Void Method with Parameters • Calling a Non-void Method • `String` Objects: Concatenation, Literals, and More • `String` Methods • Wrapper Classes: `Integer` and `Double` • Using the `Math` class

Unit 5: Boolean Expressions and `if` Statements (3 weeks/15 hours)

This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators.

Topics Covered	• Boolean Expressions • `if` Statements and Control Flow • `if-else` Statements • `else if` Statements • Compound Boolean Expressions • Equivalent Boolean Expressions • Comparing Objects

Unit 6: Computer Systems and Software Management (4-5 weeks or 20-25 hours)

Students will compare and contrast common operating systems (Windows, Linux, OS) and explain the importance of application security. Students will also learn about the hardware components that digital technology relies on, and how their code gets executed. They will investigate security options, as well as error handling techniques and implement user accounts to enforce authentication and authorization. Towards the end of fo the unit, students will work with git to demonstrate common version control operations.

Topics Covered	• Hardware • Operating Systems • Software and Applications • Software Licenses • Error Handling • Application Security • Browser Configuration • System Administration • Command Line Interface • Executing Code

	• Version Control

Unit 7: Iteration (4 weeks or 15 hours)

This unit focuses on iteration using while and for loops and introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations.

Topics Covered	• `while` loops • `for` loops • Developing Algorithms Using Strings • Nested Iteration • Informal Code Analysis

Unit 8: Writing Classes (3 weeks/15 hours)

This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. Students will build on what they learn in this unit to represent relationships between classes through hierarchies. The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this unit.

Topics Covered	• Anatomy of a Class • Constructors • Documentation with Comments • Accessor Methods • Mutator Methods • Writing Methods • Static Variables and Methods • Scope and Access • `this` Keyword • Abstract Classes and Interfaces • Ethical and Social Implications of Computing Systems

Unit 9: Data Structures (4 weeks/20 hours)

In this unit, students learn basic data structures in Java including arrays, ArrayLists, 2 dimensional arrays and HashMaps. Understanding and mastering these fundamental data structures in is crucial as they serve as the building blocks for efficient and organized data manipulation and storage, forming the backbone of many real-world applications and algorithms.

Topics Covered	• Declaring and initializing arrays • Constructing ArrayLists • Indexing into arrays/ArrayLists • Iterating over arrays/ArrayLists • Getting the length of an array/ArrayLists • ArrayIndexOutOfBoundsException • IndexOutOfBoundsException

	• Understand array variables are references to objects • Arrays/ArrayLists as parameters and return values • Inserting and deleting array/ArrayList elements • Wrapper classes - Double, Integer • Storing objects/primitives in arrays vs. ArrayLists • Numerical representations of integers • Representations of non-negative integers in different bases • Implications of finite integer bounds • The List interface • Declaring and initializing 2-D rectangular arrays • Using nested loops to iterate through 2-D arrays • row-major order • Students reminded about indices starting at 0 • Constructing, adding to, and iterating through HashMaps • Deciding which data structures to use when designing a class

Unit 10: Algorithms and Recursion (3 weeks/15 hours)

In this unit, students will be introduced to fundamental searching and sorting algorithms including sequential search, binary search, insertion sort, selection sort, and mergesort, as well as the important concept of recursion. Mastery of these fundamental searching and sorting algorithms is essential for students as they provide the fundamental techniques for efficient data retrieval, organization, and manipulation, enabling them to solve a wide range of problems in various domains.

Topics Covered	• Informal code analysis • Space and time complexity of algorithms • Search algorithms such as linear and binary search • Sorting algorithms such as selection and insertion sort • Recursion and recursive algorithms like merge sort

Unit 11: Inheritance (3 weeks or 15 hours)

Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through *inheritance*. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. These newly created classes are called *subclasses*. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a *superclass* and will then create a hierarchy by writing subclasses to extend a superclass.

Topics Covered	• Creating Subclasses and Superclasses • Writing Constructors for Subclasses • Overriding Methods • `super` Keyword

	• Abstract Classes and Interfaces • Creating References Using Inheritance Hierarchies • Polymorphism • `Object` Superclass