Curricular Requirements		Page(s)
CR1	The course teaches students to design and implement computer-based solutions to problems.	2,3-4,5,6-7,8-9
CR2a	The course teaches students to use and implement commonly used algorithms.	3-4,5,6-7,8-9
CR2b	The course teaches students to use commonly used data structures.	5,6-7,8-9
CR3	The course teaches students to select appropriate algorithms and data structures to solve problems.	5,6-7,8-9
CR4	The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.	8-9
CR5	The course teaches students to use elements of the standard Java library from the AP Java subset in Appendix A of the AP Computer Science A Course Description.	3-4,5,6-7,8-9
CR6	The course includes a structured lab component comprised of a minimum of 20 hours of hands-on lab experiences.	3-4,5,6-7,8-9
CR7	The course teaches students to recognize the ethical and social implications of computer use.	5

Course Overview

This course serves as a foundational class for future computer science and related technical majors such as engineering, physics, or chemistry. The course has a heavy emphasis on writing computer code (students are expected to write original code a during lab hours a minimum of two hours per week, vastly exceeding the 20 hour hands on lab requirement. [CR6] Course instruction prepares students for the AP Computer Science A Exam by teaching all aspects of the AP Java subset found online (https://apstudent.collegeboard.org/apcourse/ap-computer-science-a/about-the-exam/java-subset)

Further goals for the class include: the ability to rapidly learn any new computer programming language, extend their knowledge through online investigation, and most importantly, a can-do approach to problem solving.

Major Texts

Downey, Allen. Think Java: How to Think Like a Computer Scientist. 5.1.2 ed. N.p.: Green Tea, 2012. 2012. Web. 7 Sept. 2014. <http://www.greenteapress.com/thinkapjava/thinkapjava.pdf>

Course Planner

The Master Knowledge Map (MKM) is attached. This document lists all the major content areas in the AP Java Subset. Students use it to help identify the objectives of each lesson and track their progress throughout the year.

Online Resources

- Class Website: http://mchs-cs.com/apcsa/VideoTutorial/
- CodingBat.com: Practice Java coding exercises
- GridWorld Case Study Student Manual

Unit 1

Pseudo-code and the basic structures of programming

Reading and Handouts

Handouts:

- Pseudo-code

Additional Reading:

- TJ Chapter 1

Topics

- Pseudo-code
- Variables
- Operators
- Conditionals
- Loops

Objectives

- Write solutions to simple problems in pseudo-code using the four basic structures: variables, operators, conditionals, and loops
- Complete the `helloWorld` program in Java

Assessments: [CR1]

- Evaluate the student's solution of "how to navigate to class" pseudo-code
- Pseudo-code assignment: Following algorithm to calculate exponents

Unit 2

Anatomy of a Java Program, console output, Expressions, Variables, Operators, Conditional Statements, Loops, integer and floating-point evaluation, Type Conversion, and writing static methods with parameters and return values

Reading and Handouts

Handouts:

- Newtonian Square Pseudo-code
- Anatomy of a Java Program
- Expressions
- Variable Assignment Operators
- Conditional Statements
- Loop Statements
- Type Conversion
- Two Operation Types
- Introduction to Writing Methods

Additional Reading:

- TJ Ch. 2.2-2.6, p. 15-21
- TJ Ch. 3, p. 25-36
- TJ Ch. 4.1-4.7, p. 39-43
- TJ Ch. 6.1, p. 55-56
- TJ Ch. 6.5-6.7, p. 61-63
- TJ Ch. 7.1-7.2, p. 75-77

Topics

- Converting Pseudo-code to Java Structures and Syntax
- Using `System.out` with `print` and `println` to output to the console
- Primitive Data Types and String Objects
- Variables
- Variable Assignment
- Operators
- Conditionals
- Loops
- Type Conversion
- The Two Operation Types (integer and floating point math)
- Writing Methods with Parameters and Return Values

Objectives

- Given variable values, students will be able to correctly evaluate expressions that include variables, mathematical, relational, and logical operators, in the context of variable assignment, conditional, and loop statements
- Students will be able to accurately predict the flow and values as they change in a Java program.
- Write the Newtonian Square Algorithm in Java given the pseudo-code
- Students will be able to write statics methods with given inputs and expected return values

Strategies

Students will have time in class to complete and self-grade worksheets to check for understanding:

- Worksheet: Expressions – Accurately evaluate simple expressions

- Worksheet: Variable Assignment – Correctly identify type, appropriate variable names, and values
- Worksheet: Loop Statements – Accurately follow simple program with loops
- Worksheet: Type Conversion – Correctly identify the resultant type after conversion

Students will work for at least 5 hours in the computer lab to complete the lab assignments.

Assessments: [CR1][CR2a][CR5][CR6]

- Assignment: Conditional Statements with Expression Evaluation – Accurately evaluate complex expressions in variable assignments and conditional statements
- Lab: Newtonian Square Algorithm in Java – Turn pseudo-code into a complete Java program
- Assignment: Two Operation Types Assignment – Correctly evaluate the results of type conversion during expression evaluation
- Lab: Writing Methods-
    - Lab Assignment 1: Random Number in Range – Write a static method capable of generating a random number between any two parameter values
    - Lab Assignment 2: Evaluating Numbers (Even, Prime, Self-Divisor, and Palindromic) – Write static methods capable of determining if the numbers between 1 and 100 fall into specified number categories
    - Lab Assignment 2a: Categorizing Random Numbers – Write a program that generates a random number that meets a specified number category
- Exam: Unit Test

Unit 3

Loops, Nested Loops, Variable Scope, String objects and methods, ethical use of computers, and ethical hacking

Reading and Handouts

Handouts:

- For Loops
- Nested Loops
- Variable Scope
- String Methods
- String Methods: Algorithm for Converting to Uppercase

Additional Reading:

- TJ Ch. 8, p. 91-99

Topics

- For Loops
- Nested For Loops
- Variable Scope
- String Objects
- String Methods

Objectives

- Students will be able to process strings using for loops and string methods.

Strategies

Students will read the handouts regarding loops, complete a loops lab followed by a nested loops lab, and then examine variable scope in classes and methods before delving into the String class and methods. In this unit students will complete lab work (some is part of the formal assessment process). , This unit is heavily lab intensive as all the assignments include a lab component:

- Lab: Loops – Students will write several loops to calculate various sums
- Worksheet(s): String methods 1-3 – Accurately evaluate calls to the String methods in three practice sets

A brief lecture on the impact of computers and technology on the world will be followed by a class discussion of the ethical use of computers in society. A second lecture will discuss the role of hacking in computing history, the case of Kevin Mitnick, and the evolution of ethical hacking.
Students will visit the library to learn about copyrights, trademarks, fair use, and various license options for their work-product.

Students will begin using CodingBat.com to complete practice exercises:

- Warmup-1 and Warmup-2
- String-1
- Logic-1

Assessments: [CR1][CR2a][CR3][CR5][CR6][CR7]

- Lab: Nested Loops – Students will write algorithms to create various patterns in the console using nested loops
- Quiz: String Methods
- Students will write a two-page essay on either the ethical use of computers or ethical hacking

Unit 4

Arrays, Two Dimensional Arrays, Sorting with Arrays, the Selection, Insertion, and Merge Sort Algorithms

Reading and Handouts

Handouts:

- Arrays Introduction
- Working with Arrays
- Two Dimensional Arrays
- Introduction to Sorting Algorithms
- Selection Sort
- Insertion Sort
- Merge Sort
- Doing the Merge

Additional Reading:

- TJ Ch. 12, p. 149-158

Topics

- Arrays
- Two Dimensional Arrays
- Searching an Array
- Sorting Algorithms
- The Selection Sort
- The Insertion Sort
- The Merge Sort

Objectives

- Students will be able to create, iterate, and manipulate arrays and two-dimensional arrays
- Students will be able to identify the most efficient sorting algorithm based upon the data to be sorted

Strategies

This unit focuses on two topics: arrays, and sorting algorithms. Arrays are introduced and the students are given lab time to practice manipulating one and two dimensional arrays before being introduced to the common sorting algorithms and the concept of sorting efficiency. Students then gain practice manipulating arrays by writing methods to sort arrays using each of the three sorting algorithms covered in the AP subset, gathering data on the efficiency of each in a graphic organizer.

Students will continue using CodingBat.com to complete practice exercises:

- String-2
- Logic-2
- Array-1 and Array-2

Assessments: [CR1][CR2a][CR2b][CR3][CR5][CR6]

- Lab: Arrays Introduction 1 – Students will write methods to generate arrays of names by combining given arrays containing first, middle, and last names
- Lab: Arrays Introduction 2 – Students will write methods to create an array of randomly generated names with no duplicates
- Lab: Two Dimensional Arrays – Students will write an algorithm for storing names in a two dimensional array with each sub-array representing a category

- Labs: Selection, Insertion, and Merge Sort – Students will write a Java methods for sorting using each sorting algorithm that can also return a measure of efficiency, summarizing the data on the Sorting Worksheet
- Worksheet: Sorting – Students will collect data under the best, worst, and average case of each sorting algorithm, graph the average case data, and then use the data to summarize the efficiency (big o) of each

Unit 5

Objects, Inheritance, GridWorld, Interfaces, Abstract Classes, Generics and the List interface, the ArrayList class, and Recursion

Reading and Handouts

Handouts:

- Introduction to Objects
- GridWorld
- Interface class
- Abstract class
- Generics and the `List` interface
- The `ArrayList` class
- Recursion

Additional Reading:

- TJ Ch. 11, p. 131-144
- TJ Ch. 5, p. 51-53
- TJ Ch. 13, p. 165-177
- TJ Ch. 14, p. 181-189
- TJ Ch. 15, p. 193-200
- TJ Ch. 10, p. 123-129
- TJ Ch. 16, p. 203-211

Topics

- Objects:
  - Instantiation and Constructors
  - Instance Variables
  - Methods: Getters and Setters
  - Inheritance (is-a and has-a relationship)
- Interface class
- Abstract class
- Generics: The `List` interface and `ArrayList` class
- Putting it all together: GridWorld!

Objectives

- Students will learn how to define an object class, defining constructor methods, working with instance variables, and properly define getter and setter methods for an object
- Students will learn about the role of inheritance in design philosophy
- Students will learn the role of the Interface and Abstract classes
- Students will learn about the `List` interface and how to construct, and then manipulate an `ArrayList`
- Students will study a complex example using the GridWorld case study
- Students will learn about recursion and several methods for tracing recursive algorithms

Strategies

Students will begin by learning about concrete objects as user defined classes by creating several examples such as student records and exam scores and then expand their knowledge conceptually by learning about the interface and abstract class. Students will work with the `List` interface and `ArrayList` class using the concrete classes they have already created.

Students will then work through GridWorld Case Study Student Manual.

Finally, students will use the Recursion Curriculum Module to learn about recursion.

Students will continue using CodingBat.com to complete practice exercises:

- String-3
- Array-3
- AP-1
- Recursion-1 and Recursion-2

Assessments: [CR1][CR2a][CR2b][CR3][CR4][CR5][CR6]

- Lab: Abstract Class Assignment – Students will create an Abstract GridWorld class that can survive for only a limited number of steps
- Lab: ArrayList Assignment – Students create a class that extends the Abstract class in form of a bug with a limited lifespan
- Recursion Curriculum Module Formative and Summative Assesments