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

**Course Overview**

[Introduction paragraph]

**Texts**

Lewis, Loftus, and Cocking. *Java Software Solutions for AP Computer Science 3rd Edition.*

Boston, Mass. Addison-Wesley, 2011.

Teukolsky, Roselyn. Barron’s AP Computer Science A 6th Edition.

Hauppauge, New York. Barron’s Educational Series, Inc, 2013.

Eckell, Bruce. *Thinking in Java 4th Edition*.

#######

**Course Outline**

**Weeks 1-17**

|  |  |
| --- | --- |
| Weeks 1-2 | **Computational Architecture**   1. Numerical representations    1. digits & overflows    2. Number bases & conversions 2. Logic circuits and hardware [CR1]    1. Boolean logic & binary operations 3. Processors [CR1]    1. Sequential instructions and data    2. Demonstration of a “trivial processor” (add values, write to memory, read from memory) 4. Programming languages, compilers, and interpreters [CR1]    1. Operators    2. Integer math 5. Command line interface    1. Compiling and running a Java program    2. Java standard output 6. Converting algebraic expressions into Java code [CR1]   **Lab:**   1. “Compile” an algebraic expression for the “trivial processor” [CR6] 2. Java program: calculate final distance of a trajectory [CR6]   **Exercises:**   1. Logic circuit: design a 2-bit adder 2. Add feature to lab program: calculate maximum height of the trajectory 3. Extra credit: calculate height and distance at four (4) arbitrary points in time   (Need to determine what reading should go here) |
| Weeks 3-4 | Program Statements – Standard Input, Conditionals, & Iteration   1. Using standard input to assign values to variables    1. Java standard input 2. Using conditionals to add boundaries – when we can’t control the input    1. Conditional statements: if, if else    2. Nested if, else if    3. Demonstration of adding “equals” and “jump” with the “trivial processor” vocabulary 3. Formalizing iteration with loops    1. Iteration statements: while, for    2. Nested while, for   **Lab:**   1. “Compile” an if-then expression for the “trivial processor” [CR6] 2. Java program: calculate the height and distance of a trajectory at 0.1 second intervals [CR6]   **Exercises:**   1. Add feature to lab program: calculate maximum height of the trajectory 2. Extra credit: simultaneously calculate the height and distance at 0.1 second intervals of (4) four unique trajectories |
|  |  |

page2image1617536

|  |  |
| --- | --- |
|  |  |
| **Weeks 5-6** | **Program Statements – Conditional**  Software Development Process; control flow (sequential and conditional); Boolean expressions, laws, and truth tables; using conditional expressions in if, if-else, and nested if statements; and More operators (increment, decrement, compound assignment).   * Reading: *Java Software Solutions,* sections 3.0-3.4; Summary of Key Concepts (redacted). * Reading: Magpie Introduction and Activities 1-4 (APCS A Labs). * Exercises: *Java Software Solutions,* Self-Review Questions 3.1-3.11. * Exercises: Syntax Index Cards for if statements. |

CR1— The course teaches students to design and implement computer-based solutions to problems.

CR5— The course teaches students to use elements of the standard Java library from the AP Java subset

page4image2024640

|  |  |
| --- | --- |
|  | * Exercises: Boolean Expression and Decision Making Statement Worksheets. * Lab: *Java Software Solutions,* Programming Project 3.2 – Design, implement, and test a program that determines if a given year is a leap year. **[CR1]** * Lab: Magpie Activities 1-4 (APCS A Labs). **[CR4]** * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice   3.1-3.3, 3.9, 3.10; True/False 3.1-3.6, 3.8-3.9; AP-Style Multiple  Choice 3.2, 3.6.   * Test: Multiple Choice Exam. |
| **Weeks 7-9** | **Program Statements – Iteration**  Flow of control (iteration); using while and for statements; infinite and nested loops; and analysis of algorithms (informal comparisons of running times and exact calculation of statement execution counts). **[CR3]**   * Reading: *Java Software Solutions,* sections 3.5, 3.7 (expect Iterators and For Loops); Summary of Key Concepts (redacted). * Exercises: *Java Software Solutions,* Self-Review Questions 3.12-3.13 * Exercises: Syntax Index Cards for while, and for statements. * Exercises: Loop Worksheets. * Lab: *Java Software Solutions,* Programming Project 3.6 – Design,   implement, and test a program to count odd/even/zero digits. **[CR1]**   * Lab: *Java Software Solutions,* Programming Project 3.10 – Design,   implement, and test a hi-lo guessing game program. **[CR1]**   * Lab: *Java Software Solutions,* Programming Project 3.12 – Design,   implement, and test a program that prints two-dimensional patterns  of asterisks. **[CR1]**   * Lab: *Java Software Solutions,* Programming Project 3.14 – Design,   implement, and test a program that plays a Rock Paper Scissors game  with the user. **[CR1]**   * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice   3.4-3.8; True/False 3.7; AP-Style Multiple Choice 3.1, 3.3-3.5.   * Test: Multiple Choice Exam. |
| **Weeks 3-4** | **Objects & Primitive Data**  Simple data types (int, boolean, double, char); declarations (variable and constant); assignment and arithmetic expressions; console output (System.out.print/println); primitive types vs. objects; using classes to create objects; references; Java library classes (String, Integer, Double, Math, Scanner); and creating random numbers. **[CR5]**   * Reading: *Java Software Solutions,* sections 2.0-2.5, 2.7 (except Autoboxing), 2.8 (except The Random Class), 2.9; and Summary of Key Concepts (redacted). * Exercises: *Java Software Solutions,* Self-Review Questions 2.1-2.14, 2.16-2.20. * Exercises: Syntax Index Cards for types, constants (literals and symbolic), declaration, assignment, and concatenation. * Exercises: Declaration, Assignment, and Arithmetic Expression Worksheets. * Lab: Pretty Print – Implement and test a program to print a table using escape sequences. **[CR1]** * Lab: Base Convert – Implement and test a program to convert numbers from base 10 to 4-digit numbers in a chosen base 2-9. **[CR1]** * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice 2.1-2.10; True/False 2.1-2.7; AP-Style Multiple Choice 2.1-2.3.   Test: Multiple Choice Exam. |
| **Weeks 10-12** | **Writing Classes**  Anatomy of classes, constructors, and methods; declarations (class, interface, instance variable, method, and parameter); method overloading; method decomposition; object relationships; reasoning about programs (assertions, pre- and post-conditions); data abstraction and encapsulation; and designing and implementing a class.   * Reading: *Java Software Solutions,* sections 4.0-4.5; Summary of Key Concepts (redacted). * Reading: Elevens Introduction and Activity 1 (APCS A Labs). * Exercises: *Java Software Solutions,* Self-Review Questions 4.1-4.12. * Exercises: Syntax Index Cards for classes. * Lab: Elevens Activity 1 – Card Class (APCS A Labs). **[CR4]** |

CR1— The course teaches students to design and implement computer-based solutions to problems.

CR4— The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.

CR3— The course teaches students to select appropriate algorithms and data structures to solve problems.

|  |  |
| --- | --- |
|  | * Lab: Pongtastic Lab – Implement, and test three new classes that complete an OOP Pong game. **[CR1]** See http://nifty.stanford. edu/2003/pong/ * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice 4.1-4.10; True/False 4.1-4.10; AP-Style Multiple Choice 4.1-4.6. * Test: Multiple Choice Exam. |
| **Weeks 13-15** | **Enhancing Classes**  References, exceptions, and class design; == vs. equals; object parameter passing; error handling (runtime exceptions, throwing runtime exceptions); interfaces and abstract classes; Java library classes (Comparable and List interfaces) **[CR5]**; and identifying reusable components from existing code using classes and class libraries.   * Reading: *Java Software Solutions,* sections 5.0-5.3 (except Iterator and ListIterator Interfaces); Summary of Key Concepts (redacted). * Exercises: *Java Software Solutions,* Self-Review Questions 5.1-5.6. * Exercises: Parameter Passing Worksheet. * Lab: *Java Software Solutions,* Programming Project 5.2 – Modify an   existing Rational class to change its definition of equals and to  make it Comparable; test the modified Rational class. **[CR1]**   * Lab: *Java Software Solutions,* Programming Project 5.6 – Design and   implement a Lockable interface; modify a Coin class to make it  Lockable; test the modified Coin class. **[CR1]**   * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice   5.1-5.10; True/False 5.1-5.10; AP-Style Multiple Choice 5.1-5.6.   * Test: Multiple Choice Exam. |
| **Week 16** | **Semester Review** |
| **Week 17** | **Semester Finals** |

**Weeks 18-38**

CR1— The course teaches students to design and implement computer-based solutions to problems.

CR5— The course teaches students to use elements of the standard Java library from the AP Java subset

CR2a— The course teaches students to use and implement commonly used algorithms.

|  |  |
| --- | --- |
| **Weeks 18-21** | **1D Arrays / 2D arrays / Searching [CR2a] [CR2b]**  One- and two-dimensional arrays (creation, insertions, deletions, traversals, algorithms); searching algorithms and comparison (sequential and binary); and choosing appropriate data representation and algorithms.   * Reading: *Java Software Solutions,* sections 6.0-6.2, 6.6; Summary of Key Concepts (redacted). * Reading: PictureLab Introduction and Activities 1-9 (APCS A Labs). * Reading: CodingBat Java Arrays and Loops at http://codingbat.com/   doc/java-array-loops.html.   * Exercises: *Java Software Solutions,* Self-Review Questions 6.1-6.9. * Exercises: Array Worksheets. * Exercises: Create Working Solutions for 10 CodingBat Array-2   Problems at http://codingbat.com/java/Array-2. |

CR2b— The course teaches students to use commonly used data structures.

|  |  |
| --- | --- |
|  | * Lab: *Java Software Solutions,* Programming Project 6.4 – Design, implement, and test a program that inputs integers and produces a histogram. **[CR1]** * Lab: Picture Lab Activities 1-9 (APCS A Labs). **[CR4]** * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice   6.1-6.5, 6.8; True/False 6.1-6.7; AP-Style Multiple Choice 6.1-6.5.   * Test: Multiple Choice Exam. |
| **Weeks 22-24** | **Lists / ArrayLists / Selection and Insertion Sorts**  Lists and ArrayLists (creation, insertions, deletions, traversals, algorithms); **[CR2b] [CR5]** sorting algorithms and comparison (selection and insertion) **[CR2a] [CR3]**; and choosing appropriate data representation and algorithms. **[CR3]**   * Reading: *Java Software Solutions,* sections 6.3-6.4, 6.7; Summary of Key Concepts (redacted). * Reading: Elevens Activities 2-4 (APCS A Labs). * Exercises: *Java Software Solutions,* Self-Review Questions 6.10, 6.12. * Exercises: List/ArrayList Worksheets. * Lab: Elevens Activities 2-4 – Deck Class (APCS A Labs). **[CR4]** * Review: List Algorithms Worksheet. * Test: Multiple Choice Exam. |
| **Weeks 25-27** | **Inheritance**  Inheritance (subclasses, overriding, hierarchies, using class members, polymorphism, and class hierarchy design); interfaces and abstract classes; Java library classes (Object) **[CR5]**; reading and understanding class specifications and relationships among classes (“is-a” and “has-a”); understanding and implementing a given class hierarchy; extending a given class using inheritance; and applying functional decomposition.   * Reading: *Java Software Solutions,* sections 7.0-7.7; Summary of Key   Concepts (redacted).   * Reading: Elevens Activities 6-9 (APCS A Labs). **[CR4]** * Exercises: *Java Software Solutions,* Self-Review Questions 7.1-7.12. * Lab: Elevens Activities 6-9 – Board and AbstractBoard Classes   (APCS A Labs).   * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice   7.1-10; True/False 7.1-10; AP-Style Multiple Choice 7.1-7.6.   * Test: Multiple Choice Exam. |
| **Weeks 28-30** | **Recursion / Merge and Quick Sorts**  Recursive thinking, programming, and sorting; flow of control (recursion); sorting algorithms (merge **[CR2a]** and quick) and comparison with other sorts. **[CR3]**   * Reading: *Java Software Solutions,* sections 8.0-8.3; Summary of Key   Concepts (redacted).   * Exercises: *Java Software Solutions,* Self-Review Questions 8.1-8.9 * Exercises: Tracing Recursion Worksheet. |

CR1— The course teaches students to design and implement computer-based solutions to problems.

CR4— The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.

CR2b— The course teaches students to use commonly used data structures.

CR5— The course teaches students to use elements of the standard Java library from the AP Java subset

CR2a— The course teaches students to use and implement commonly used algorithms.

CR3— The course teaches students to select appropriate algorithms and data structures to solve problems.

|  |  |
| --- | --- |
|  | * Lab: Numbrix – Implement and test an OOP recursive program which solves Numbrix puzzles. **[CR1]** See http://www.parade.com/numbrix * Review: Clicker Questions – *Java Software Solutions,* Multiple Choice 8.1-8.10; True/False 8.1-8.10; AP-Style Multiple Choice 8.1-8.6. * Test: Multiple Choice Exam. |
| **Weeks 31-33** | **AP Test Practice Exam / AP Review**  AP Computer Science A Examination (practice, content, materials, timing, tips).   * Reading: *Barron’s AP Computer Science A,* Chapters 1-8. * Exercises: *Barron’s AP Computer Science A,* Chapters 1-8 Multiple-   Choice Questions.   * Exercises: Create Working Solutions for all 21 CodingBat AP-1.   Problems at http://codingbat.com/java/AP-1   * Test: AP Practice Examination. * Quizzes: Daily 3-4 question multiple-choice quizzes (questions from   *Barron’s AP Computer Science A,* Multiple-Choice Questions). |
| **Week 34** | **Ethical and Social Implications of Computer Use; AP Examination**  Responsible use of computer systems (system reliability, privacy, intellectual property, legal issues, and social and ethical ramifications of computer use). **[CR7]**   * Reading: one student-chosen chapter of *Blown to Bits*. * Assignment: Prepare a one-page summary of the chapter and   participate in a classroom discussion of it. **[CR7]** |
| **Weeks 35-37** | **Post-AP Project – RoboCode**  Cooperative programming; research; reading code; and comparing strategies and algorithms.   * Reading: Robocode website – http://robocode.sourceforge.net/ * Lab: RoboWarrior – Work in pairs to design, implement, and test a   competitive Robocode robot. |
| **Week 38** | **Semester Finals** |