

# Course Syllabus

## Description:

Students must take the Advanced Placement Exam in order to receive Advanced Placement credit.

The AP Computer Science A course is equivalent to the first segment of a college level computer science course. The course involves developing the skills to write programs or part of programs to correctly solve specific problems. AP® Computer Science A also emphasizes the design issues that make programs understandable, adaptable, and when appropriate, reusable. At the same time, the development of useful computer programs and classes is used as a context for introducing other important concepts in computer science, including the development and analysis of algorithms, the development and use of fundamental data structures, and the study of standard algorithms and typical applications. In addition an understanding of the basic hardware and software components of computer systems and the responsible use of these systems are integral parts of the course.

**Estimated Completion Time:** 2 segments / 32-36 weeks

## Major Topics and Concepts:

### Segment I:

#### Module One: Primitive Types

- 01.00 Orientation
- 01.01 Course Folder Management
- 01.02 Installing Java
- 01.03 Installing the BlueJ IDE
- 01.04 Student Contact Card
- 01.05 Stylish Java
- 01.06 Order of Operations
- 01.07 Printing Arithmetic Expressions
- 01.08 Primitive Data Type: ints
- 01.09 Primitive Data Type: doubles
- 01.10 Arithmetic Expressions
- 01.11 Primitive Data Type Conversions
- 01.12 Pitfalls, Surprises, and Shortcuts
- 01.13 Challenge Program
- 01.14 Discussion-Based Assessment
- 01.15 Checkpoint Beta

## **Module Two: Using Objects**

- 02.00 Introduction
- 02.01 Pseudocode, Recipe for Success
- 02.02 The Java API
- 02.03 Java's Math Class
- 02.04 Primitive Data Types: char
- 02.05 String Objects Lite
- 02.06 String Class Methods: The Basics
- 02.07 Scanner Class Methods

- 02.08 Parsing
- 02.09 Calling Methods in Java
- 02.10 Challenge Program
- 02.11 Module Two Exam

### **Module Three: Boolean Expressions and if Statements**

- 03.00 Getting Started with if Statements
- 03.01 Number Systems
- 03.02 Primitive Data Types: booleans
- 03.03 Condition Statements: if
- 03.04 Condition Statements: if-else
- 03.05 Condition Statements: if-else-if
- 03.06 Comparing Strings
- 03.07 Logical Operator
- 03.08 Discussion-Based Assessment
- 03.09 Checkpoint Delta

### **Module Four: Iteration**

- 04.00 Loops
- 04.01 While Loops (Part 1)
- 04.02 While Loops (Part 2)
- 04.03 Reading Text Files
- 04.04 For Loops
- 04.05 Nested Loops
- 04.06 Writing Text Files
- 04.07 Challenge Program

- 04.08 Module Four Exam

## **Module Five: Writing Classes**

- 05.00 Procedural and OOP Styles
- 05.01 Defining New Static Methods: Part 1
- 05.02 Defining New Static Methods: Part 2
- 05.03 Getting Started with Objects
- 05.04 Instances of a Class
- 05.05 Default Constructors
- 05.06 Constructors with Parameters
- 05.07 Overloading Methods and Using Two Classes
- 05.08 Constructing Multiple Objects
- 05.09 Programming Ethics and Security
- 05.10 Discussion-Based Assessment
- 05.11 Part A Segment One Exam - Multiple-Choice
- 05.11 Part B Segment One Exam - Free Response

## **Segment II**

### **Module Six: Arrays**

- 06.00 Getting Started With Arrays
- 06.01 One-Dimensional Arrays
- 06.02 Formatting Output
- 06.03 The for-each Loop
- 06.04 Arrays of Objects
- 06.05 Computer Science Lab: Magpie Part 1

- 06.06 Discussion-Based Assessment
- 06.07 Part 1 Challenge Exam - Multiple-Choice
- 06.07 Part 2 Challenge Exam - Free Response

### **Module Seven: ArrayList**

- 07.00 Introduction to Standard Algorithms
- 07.01 ArrayLists I
- 07.02 ArrayLists II
- 07.03 Challenge Program
- 07.04 Traversals Lesson
- 07.05 Replacements Lesson
- 07.06 Insertions Lesson
- 07.07 Deletions Lesson
- 07.08 Challenge Program
- 07.09 Insertion Sort
- 07.10 Selection Sort
- 07.11 Computer Science Lab: Magpie Part 2
- 07.12 Checklist

### **Module Eight: 2D Array**

- 08.00 Introduction to Searching
- 08.01 Sequential Search
- 08.02 Binary Search
- 08.03 Challenge Program
- 08.04 2D Array Part 1

- 08.05 2D Array Part 2
- 08.06 Computer Science Lab: Picture
- 08.07 Computer Science Lab: Picture
- 08.08 Discussion-Based Assessment
- 08.09 Checklist

## **Module Nine: Inheritance**

- 09.00 Introduction to Inheritance and Polymorphism
- 09.01 Extending Classes
- 09.02 Class Hierarchies
- 09.03 Polymorphism
- 09.04 Overriding Methods
- 09.05 Design Strategy: Iterative and Incremental
- 09.06 Static Means Never Having to Instantiate an Object
- 09.07 Class Variables and Constants
- 09.08 Thinking Outside the Box
- 09.09 Part 1 Challenge Exam - Multiple-Choice
- 09.09 Part 2 Challenge Exam - Free Response
- 09.10 Checkpoint

## **Module Ten: Recursion**

- 10.00 Getting Started with Recursion
- 10.01 Divide et Impera
- 10.02 Real-World Recursion
- 10.03 The Recursive Leap of Faith

- 10.04 There and Back Again
- 10.05 Are We There Yet?
- 10.06 Recursive Strings and Things
- 10.07 Challenge Program
- 10.08 Computer Science Lab: Elevens
- 10.09 AP Exam Review
- 10.10 Practice Exams
- 10.11 Discussion-Based Assessment
- 10.12 Part A Segment Two Exam - Multiple-Choice
- 10.12 Part B Segment Two Exam - Free Response

### **Course Assessment and Participation Requirements:**

To achieve success, students are expected to submit work in each course weekly. Students can learn at their own pace; however, "any pace" still means that students must make progress in the course every week. To measure learning, students complete self-checks, practice lessons, multiple choice questions, projects, discussion-based assessments, and discussions. Students are expected to maintain regular contact with teachers; the minimum requirement is monthly. When teachers, students, and parents work together, students are successful.

---

