Cornwall-Lebanon School District Curriculum Overview

AP Computer Science A- High School

33 length of time in weeks	Concepts & Competencies	Common Assessments	Academic Standards (PA Core if applicable)
Unit 1 — 5	<u>Java Basics</u> A. Student will be able to write algorithms that include sequential, conditional and iterative control. B. Student will be able to analyze algorithms through statement execution counts and information run-time comparisons. C. Student will be able to analyze problems and develop potential solutions using RAD and pseudocode. D. Student will be able to use primitive data types int, double and Boolean. E. Student will be able to use final keyword for final block scope constants and static final class scope constants. F. Student will be able to use arithmetic operators: +, -, *, /, and %. G. Student will be able to use assignment operators: =, +=, -=, *=, /= and %=. H. Student will be able to use the postfix form of the increment/decrement operators ++ and --. I. Student will be able to use /* */, and // comments. J. Student will be able to use numeric casts (int) and (double) and understand "truncation towards 0" behavior as well as the fact that positive floating-point numbers can be rounded to the nearest integer as (int)(x + 0.5), negative numbers as (int)(x - 0.5). K. Student will be able to use System.out.print and System.out.println for program output. L. Student will be able to recognize and fix compile-time, run-time and logic errors using a debugger, adding extra output statements or hand-tracing code.	➢ Programming Project Completion ➢ Unit 1 Test	APCS A Subset

	M. Student will be able to convert numerical representations of integers between different number bases (binary, octal, hexadecimal). N. Student will be able to use reference types such as the String class and its associated methods: length(), equals(other), substring (to, from), substring (from), indexOf(str), compareTo(other) as well as String concatenation, conversion of numbers to strings and invoking of toString on objects. O. Student will be able to use escape sequences inside strings \\, \", \n. P. Student will be able to use relational operators (==, !=, <, , >=) and logical operators (&&, ||, !) including the "short circuit" evaluation of the && and || operators. Q. Student will be able to use control structures if, if/else, ~~while, for, enhanced for (for-each) and return.~~ R. Student will be able to use control structures ~~if, if/else~~, while, for, enhanced for (for-each) and return. S. Student will be able to create, fill and traverse one-dimensional arrays of both primitive types (e.g., int[]) and objects (e.g., String[]) to include initialization of named arrays (int[] arr = { 1, 2, 3 };).		
Unit 2 6	<u>Advanced Java</u> A. Student will be able to use the Math class and its associated methods: abs(int x), abs(double x), pow(base, exponent), sqrt(double x) and random(). B. Student will be able to use the Integer class, its constructor, its intValue() method and its MIN_VALUE and MAX_VALUE constants. C. Student will be able to use the Double class, its constructor and its doubleValue() method. D. Student will understand the difference between object equality (equals) and identity (==). E. Student will be able to understand the exceptions that occur when their programs contain errors (in particular, NullPointerException, ArrayIndexOutOfBoundsException, ArithmeticException, ClassCastException, IllegalArgumentException).	➢ Programming Project Completion ➢ Unit 2 Test	APCS A Subset

	F. Student will be able to use the List interface to create and use ArrayLists and its associated methods: size(), add(obj), add(index, obj), get(index), set(index, obj), and remove(index). G. Student will be able to create, fill and traverse two-dimensional including understanding that arr[0].length is the number of columns in a rectangular two-dimensional array. H. Student will be able to understand the use of recursion and trace through code that applies it. I. Student will be able to analyze the logic behind programming code and identify correct outcomes. J. Student will be able to provide an overview of the key steps and considerations related to software engineering and design. K. Student will be able to use both top-down and bottom-up implementation techniques. L. Student will be able to develop program methods using procedural abstraction. M. Student will be able to develop appropriate test cases including boundary cases and perform both unit and integration testing. N. Student will be able to identify program correctness through pre- and post-condition and assertions.		
Unit 3 6	<u>Object Oriented Java</u> A. Student will be able to write a class definition when given a general description of the class. B. Student will be able to identify and use method overloading (e.g. MyClass.method(String str) and MyClass.method(int num)) and understand that the signature of a method depends only on the number, types, and order of the parameters, and not on its return type. C. Student will be able to use visibility modifiers such as public classes and private instance variables, and public or private for methods, constructors and constants (static final variables). D. Student will be able to implement constructors that initialize all instance variables. E. Student will be able to describe the connection between OOP and encapsulation/information hiding. F. Student will be able to construct objects with the new operator, to supply construction parameters, and to invoke accessor and modifier methods as well as modify existing	➢ Programming Project Completion ➢ Unit 3 Test	APCS A Subset

	classes (by adding or modifying methods and instance variables) and designing new classes. G. Student will be able to use static methods appropriately by invoking them only through a class, never an object (i.e., ClassName.method(), not obj.method()). H. Student will be able to use static final variables. I. Student will be able to assign and check for null references. J. Student will be able to use the keyword “this” to pass the implicit parameter in its entirety to another method (e.g., obj.method(this)) and for descriptions such as "the implicit parameter this". K. Student will be able to use basic code packages and have a reading knowledge of import statements of the form: import packageName.subpackageName.ClassName;		
Unit 4 9	<u>Inheritance</u> A. Student will understand inheritance hierarchies. B. Student will be able to extend classes and have a knowledge of inheritance that includes understanding the concepts of method overriding and polymorphism and implementation of their own subclasses. C. Student will be able to use the keyword “super” to invoke a superclass constructor (super(args)) or to invoke a superclass method (i.e., super.method(args)). D. Student will understand that conversion from a subclass reference to a superclass reference is legal and does not require a cast and that class casts (generally from Object to another class) are part of the AP Java subset, to enable the use of generic collections, for example: Person p = (Person)people.get(i); E. Student will be able to define his/her own abstract class and read the definitions of abstract classes and understand that the abstract methods need to be redefined in non-abstract classes.	➢ Programming Project Completion ➢ Unit 4 Test ➢ Successful Completion of 10-hour Lab	APCS A Subset
Unit 5 7	<u>Interfaces</u> A. Student will be able to design, create and modify an interface. B. Student will be able to design, create and modify classes that implement interfaces. C. Student will be able to identify and use interfaces to create polymorphic behavior.	➢ Programming Project Completion ➢ Unit 5 Test ➢ Successful Completion of 10-hour Lab	APCS A Subset