AP Computer Science syllabus correlation to *Introduction to Programming with Java*

This summary compares the AP Compute Science A/AB syllabus to Outsource Laboratories Press text, *Introduction to Programming with Java*. The table lists all the topics covered in the "A" and "AB" or the "AB only" syllabus. It details where the topic is covered and if a hands-on lab exercise is included. It also lists extended topics as potential enhancements to the curriculum.

Simple Language Features and Constructs					
Topic	**A, AB**	**AB Only**	**Potentially Relevant to CS1 /CS2**	**OL302 Reference** (Unit: page)	**Example & exercise**
Primitive Data Types	int, double, Boolean		`short, long, byte, char, float`	Unit 3: 12	Example & exercise
Arithmetic Operators	`+ , -, *, /, %, ++, --`		Using the values of `++, --` expressions in other expressions	Unit 3: 45, 63	Example & exercise
Assignment Operators	`=, +=, -=, *=, /=, %=`			Unit 3: 61	Example & exercise
Relational Operators	`==, !=, <, <=, >, >=`			Unit 4: 4	Example & exercise
Logical Operators	`&&, ||, !` and short-circuit evaluation		`<<, >>, >>>, &, ~, |, ^, ?:`	Unit 4: 6, 7	Example & exercise
Primitive Data Type Casting	`(int), (double)`		Other numeric casts such as (`char`) or (`float`)	Unit 3: 43	Example & exercise
Strings	String concatenation, Conversion of numbers to String by using toString() on objects		`StringBuffer`, processing string input, parsing using `StringTokenizer`	Unit 3: 32,Unit 7:9	Example & exercise
	Escape sequences `\", \\, \n` inside strings		Other escape sequences `(\', \t, \unnnn)`	Unit 7: 3,4	Example & exercise
I/O	`System.out.print, System.out.println`		`System.in`, Stream input/output, GUI input/output, parsing input, formatted output	Unit 3: 19, Throughout	Example & exercise
	Input mechanism adhoc			Unit 3: 15	Example & exercise
Arrays	1-Dimensional	2-Dimensional rectangular arrays	Arrays with 3 or more dimensions, ragged arrays	Unit 8: 1-31	Example & exercise
Control Flow	`if, if/else, while, for, return`		`do/while, switch, break, labeled break, continue`	Unit 4, Unit 5	Example & exercise
Comments	`/* */, //`		`Javadoc comments /**`	Unit 2: 20	Example

Object Oriented Programming Features					
Topic	**A, AB**	**AB Only**	**Potentially Relevant to CS1 /CS2**	**OL302 Reference** (Unit: page)	**Example & exercise**
OO Concepts	Classes, inheritance, polymorphism, encapsulation			Unit 9: 3-14 Unit 11: 29-32	Example & exercise
Classes	Class design and implementation			Unit 9	Example & exercise
	Understand the concepts of abstract classes and interfaces.			Unit 9:15-16	Example & exercise
	Design and implement subclasses, modify interface and subclass implementations			Unit 9: 17-19 Unit 11: 1-53	Example & exercise
	Class variables and methods			Unit 11: 8-23	Example & exercise
	write interface and class declarations from problem description	define own interfaces and classes		Unit 9: 12, 27 Unit 11:40-41	Example & exercise
Visibility	Classes: `public` Instance Variables: `private` Methods: `public` or `private`		`protected, package`	Unit 10: 9 Unit 11:45-47	Example
Implementing Methods	method signature, method overloading			Unit 6: 1-22 Unit 10: 17 Unit 11: 12,	Example & exercise
Objects from classes	use of `new` operator to create objects			Unit 10: 8,	Example
	Constructors and constructor parameters			Unit 11:24-28	Example & exercise
	initialize instance variables		default initialization of instance variables, initialization blocks	Unit 10: 15	
	class constants are initialized with an initializer, e.g., `public static final int MAX_SCORE = 5;`			Unit 3:44	
	Accessor/getter and modifier/setter methods			Unit 9:10	Example & exercise
`Final` keyword	block scope constants, class scope constants		final parameters, instance variables, methods, classes	Unit 3:44	Example
`Static` keyword	static final variables, static methods, invoking through a class, when to use static methods			Unit 11	Example

Object Oriented Programming Features (continued)					
Topic	**A, AB**	**AB Only**	**Potentially Relevant to CS1 /CS2**	**OL302 Reference** (Unit: page)	**Example & exercise**
Object Comparison	Object identity (==) and object equality (equals)		clone, implementation of equals	Unit 11:35	Example
Object casting	subclass to superclass, object to class casting, e.g., `Person p = (Person) people.get(i);`		`Instanceof`	Unit 11:44	Example
Object reference	`null` reference			Unit 10: 25	Example
`This` keyword	restricted to passing the implicit parameter in its entirety to another method, e.g., `obj.method(this)`		`this.var=var`, `this.method(args)`, `this(args)`	Unit 10: 13	Example
`super` keyword	to invoke a super class contructor (super(args)) and invoke a super class method, e.g., `super.method(args)`			Unit 11:33	Example
Class Nesting			nested classes, inner classes		
Threading			multi-threading		

Program Implementation					
Topic	**A, AB**	**AB Only**	**Potentially Relevant to CS1 /CS2**	**OL302 Reference** (Unit: page)	**Example & exercise**
Exceptions	understand simple exceptions- `NullPointerException`, `ArrayIndexOutOfBoundsException`, `ArithmeticException`, `ClassCastException`, `IllegalArgumentException`	throwing standard unchecked exceptions- `IllegalStateException`, `NoSuchElementException` for ADT implementation	Checked exceptions, `try/catch/finally`, `throws` modifier	Unit 14:1-23 Unit 8: 15	Example & exercise
Packages	understanding packages, import statement, e.g., import packageName.subpackageName.ClassName;		import packageName.subpackageName.*;	Unit 3:15	Example
Main Method	very simple for case studies only		program invocation through main method, command-line arguments	Unit 2:19-20	Example

Using Standard Classes and Interfaces With Their Required Methods			
Classes/Interfaces	**Methods**	**OL302 Reference** (Unit: page)	**Example & exercise**

`class java.lang.String implements java.lang.Comparable`	• `int compareTo(Object other)`
	// specified by java.lang.Comparable
	• `boolean equals(Object other)`
	• `int length()`
	• `String substring(int from, int to)`
	// returns the substring beginning at `from`
	// and ending at `to-1`
	• `String substring(int from)`
	//returns substring(from, length())
	• `int indexOf(String s)`
	// returns the index of the first occurrence of `s;`
	// returns `-1` if not found
		Unit 7: 11-21	Example

`Interface java.lang.Comparable`	• `int compareTo(Object other)`
	// return value < 0 if `this` is less than `other`
	// return value = 0 if `this` is equal to `other`
	// return value > 0 if `this` is greater than `other`
		Unit 11: 11-40,42	Example

`class java.lang.Object`	• boolean equals(Object other) • String toString() • int hashCode()	Unit 11: 11-35	Example

`class java.util.ArrayList implements java.util.List`	Methods in addition to the `List` methods:
	• `void add(int index, Object x)`
	// inserts `x` at position `index`, sliding elements
	// at position `index` and higher to the right
	// (adds 1 to their indices) and adjusts `size`
	• `Object remove(int index)`
	// removes element from position `index`, sliding elements
	// at position `index + 1` and higher to the left
	// (subtracts 1 from their indices) and adjusts `size`
		Unit 13: 55-58	Example & exercise

`class java.util.TreeSet implements java.util.Set`		Unit 13: 67-69	Example & exercise
`class java.util.LinkedList implements java.util.List`	• Methods in addition to the `List` methods • `void addFirst(Object x)` • `void addLast(Object x)` • `Object getFirst()` • `Object getLast()` • `Object removeFirst()` • `Object removeLast()`	Unit 13: 65-68	Example & exercise

Using Standard Classes and Interfaces With Their Required Methods (continued)			
Classes/Interfaces	**Methods**	**OL302 Reference** (Unit: page)	**Example & exercise**
`interface java.util.Set`	• `boolean add(Object x)` • `boolean contains(Object x)` • `boolean remove(Object x)` • `int size()` • `Iterator iterator()`	Unit 13: 69	Example & exercise (TreeSet)
`interface java.util.Map`	• `Object put(Object key, Object value)` • `Object get(Object key)` • `boolean containsKey(Object key)` • `int size()` • `Set keySet()`	Unit 13: 73	Example & exercise (HashMap & Hashtable)
`interface java.util.ListIterator extends java.util.Iterator`	• Methods in addition to the `Iterator` methods • `void add(Object x)` • `void set(Object x)`		No example/exercise

`interface java.util.List`	• boolean add(Object x)
	• int size()
	• Object get(int index)
	• Object set(int index, Object x)
	// replaces the element at index with x
	// returns the element formerly at the specified position
	• Iterator iterator()
	• ListIterator listIterator()
		Unit 13: 56	Example & exercise (ArrayList)

`interface java.util.Iterator`	• `boolean hasNext()` • `Object next()` • `void remove()`	Unit 13: 82	Example
`class java.util.TreeMap implements java.util.Map`		Unit 13: 76	Example & exercise (HashMap & Hashtable)
`class java.util.Random`	• `int nextInt(n)` // returns an integer in the range from `0` to `n-1` inclusive • double nextDouble()	Unit 11: 49, 14:12	Example

Using Standard Classes and Interfaces With Their Required Methods (continued)			
Classes/Interfaces	**Methods**	**OL302 Reference** (Unit: page)	**Example & exercise**
`class java.util.HashSet implements java.util.Set`		Unit 13: 69	Example & exercise (TreeSet)
`class java.util.HashMap implements java.util.Map`		Unit 13: 73	Example & exercise
`class java.lang.Math`	• static int abs(int x) • static double abs(double x) • static double pow(double base, double exponent) • static double sqrt(double x)	Unit 11: 49	No example/exercise

`class java.lang.Integer implements java.lang.Comparable`	• Integer(int value)
	// constructor
	• int intValue()
	• boolean equals(Object other)
	• String toString()
	• int compareTo(Object other)
	// specified by java.lang.Comparable
		Unit 13:60	No example/exercise

`class java.lang.Double implements java.lang.Comparable`	• Double(double value)
	// constructor
	• double doubleValue()
	• boolean equals(Object other)
	• String toString()
	• int compareTo(Object other)
	// specified by java.lang.Comparable
		Unit 13:60	No example/exercise