JAVA SUBSET FOR AP
CS AP EXAM - May 17, 2019 (pm)

Will be tested on the exam	Potentially relevant but NOT specifically tested
1. The primitive types int, double, and boolean.	The other primitive types short, long, byte, char, and float /
2.The increment/decrement operators ++ and -- .	The ternary ?: operator
3.The assignment operator = is part of the AP Java subset. The combined arithmetic/assignment operators +=, -=, *=, /=, %= are part of the AP Java subset although they are used simply as a shorthand and will not be used in the adjustment part of a for loop.	
4.Relational operators ==, !=, <, <=, >, >=	
5.Logical operations &&, ||, ! . Students need to understand the "short circuit" evaluation of the && and || operators.	The logical &, | and ^ and the bit operators <<, >>, >>>, &, ~, |, ^
6. The numeric casts (int) and (double) are part of the AP Java subset. Students are expected to 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).	Autoboxing, that is the automatic conversion between primitive types (int, double) and the corresponding wrapper classes (Integer, Double)
7. String concatenation + is part of the AP Java subset. Students are expected to know that concatenation converts numbers to strings and invokes toString on objects.	the StringBufferclass
8.The escape sequences inside strings \\, \", \n are part of the AP Java subset	The \t escape and Unicode \uxxxx escapes are not in the subset. The\' escape is only necessary inside character literals.

Will be tested on the exam	Potentially relevant but NOT specifically tested
9.if reading input is necessary, it will be indicated in a way similar to double x = call to a method that reads a floating-point number; or double x = IO.readDouble(); // read user input	I/O can be done in many different ways. Because of this,specific I/O features are not tested on the AP CS (System class and Scanner)
10.System.out.print and System.out.println	Converting strings to numeric values (e.g., with Integer.parseInt) Formatted output (e.g., with NumberFormat or System.out.printf)
11.In case studies, program invocation with main may occur, but the main method will be kept very simple.	
12.Arrays: one-dimensional arrays and two-dimensional rectangular arrays are part of the AP Java subset. Both arrays of primitive types (e.g., int[]) and arrays of objects (e.g., Student[]) are in the subset. Initialization of named arrays (int[] arr = { 1, 2, 3 };)	Arrays with more than two dimensions (e.g., rubik = new Color[3][3][3]) are not in the subset. "Ragged" arrays (e.g., new int[3][]) .Students need to know that an int[3][3] really is an "array of arrays" whose rows can be replaced with other int[] arrays. Students are also expected to know that arr[0].length is the number of columns in a rectangular two-dimensional array arr. Anonymous arrays (e.g., new int[] { 1, 2, 3 }).
13.The control structures if, if/else, while, for, (including the enhanced for loop), return	The do/while, switch, plain and labeled break and continue statements
14. Method overloading (e.g., MyClass.method(String str) and MyClass.method(int num))	
15. Classes: Students are expected to construct objects with the new operator, to supply construction parameters, and to invoke accessor and modifier methods. Students are expected to modify existing classes (by adding or modifying methods and instance variables). Students are expected to design their own classes.	
16. Visibility: In the AP Java subset, all classes are public. All instance variables are private. Methods, constructors, and constants (static final variables) are either public, or private.	The AP Java subset does not use protected and package (default) visibility.

Will be tested on the exam	Potentially relevant but NOT specifically tested
17.The AP Java subset uses /* */, // and /** */ comments. Javadoc comments@param and @return, are part of the subset.	
18. The final keyword is only used for final block scope constants and static final class scope constants. final parameters or instance variables	Final methods and final classes are not in the subset.
19.The concept of static methods is a part of the subset. Students are required to understand when the use of static methods is appropriate. In the exam, static methods are always invoked through a class, never an object (i.e., ClassName.method(), not obj.method()).	
20. Static variables are part of the subset. The null reference is part of the AP Java subset	
21. The use of this is restricted to passing the implicit parameter in its entirety to another method (e.g., obj.method(this)) and to descriptions such as "the implicit parameter this".	Using this.var or this.method(args) is not in the subset. In particular, students are not required to know the idiom "this.var = var", where var is both the name of an instance variable and a parameter variable. Calling other constructors from a constructor with the this(args)notation
22. Students are expected to implement constructors that initialize all instance variables. Class constants are initialized with an initializer: public static final int MAX_SCORE = 5;	The rules for default initialization (with 0, false or null) are not in the subset. Initializing instance variables with an initializer and initialization blocks.
23.Students are expected to extend classes and implement interfaces. Students are also expected to have a knowledge of inheritance that includes understanding the concepts of method overriding and polymorphism. Students are expected to implement their own subclasses.	
24.Students are expected to read the definitions of interfaces and abstract classes and understand that the abstract methods need to be implemented in a subclass. Students are expected to write interfaces or class declarations when given a general description of the interface or class.	

Will be tested on the exam	Potentially relevant but NOT specifically tested
25. Students are expected to understand the difference between object equality (equals) and identity (==).	The implementation of equals and hashCodemethods
26. Students are expected to understand that conversion from a subclass reference to a superclass reference is legal and does not require a cast. 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);	The instanceof operator is not in the subset. Array type compatibility and casts between array types
27. Students are expected to have a basic understanding of packages and a reading knowledge of import statements of the form import packageName.subpackageName.ClassName;	import statements with a trailing *, packages and methods for locating class files (e.g., through a class path)
28. The use of generic collection classes and interfaces are in the subset, but students need not implement generic classes or methods	Nested and inner classes
	Enumerations, annotations, and threads
29. Students are expected to understand the exceptions that occur when their programs contain errors (in particular, NullPointerException, ArrayIndexOutOfBoundsException, ArithmeticException, ClassCastException, IllegalArgumentException). On the AB exam, students are expected to be able to throw the uncheckedIllegalStateException and NoSuchElementException in their own methods (principally when implementing collection ADTs).	Checked exceptions are not in the subset. In particular, the try/catch/finally statements and the throws modifier

Short reference

Tested in A exam	**Potentially relevant to CS1/CS2 course but not tested on the A exam**
int, double, boolean Integer.MAX_VALUE, Integer.MIN_VALUE	short, long, byte, char, float
+ , -, *, /, %, ++, --	Using the values of ++, --expressions in other expressions
=, +=, -=, *=, /=, %=	
==, !=, <, <=, >, >=	
&&, ||, ! and short-circuit evaluation	<<, >>, >>>, &, ~, |, ^, ?:
(int), (double)	Other numeric casts such as (char) or (float)
String concatenation	StringBuffer
Escape sequences \", \\, \n inside strings	Other escape sequences (\', \t, \unnnn)
System.out.print, System.out.println	Scanner, System.in, Stream input/output, GUI input/output, parsing input, formatted output
	public static void main(String[] args)
1-dimensional arrays 2-dimensional rectangular arrays;	Arrays with 3 or more dimensions, ragged arrays
if, if/else, while, for, enhanced for, return	do/while, switch, break, continue
Modify existing classes, design classes	
public classes, private instance variables,public or private methods or constants	protected or package visibility

@param, @return	javadoc
static class variables	final local variables, finalparameters, final instance variables, final methods, finalclasses
static methods	
null, this, super, super.method(args)	this.var, this.method(args), this(args)
Constructors and initialization of static variables	Default initialization of instance variables, initialization blocks
Understand inheritance hierarchies. Design and implement subclasses. Modify subclass implementations and implementations of interfaces.	
Understand the concepts of abstract classes and interfaces.	
Understand equals, ==, and != comparison of objects String.compareTo	clone, implementation of equals, generic Comparable<T>
Conversion to supertypes and (Subtype) casts	instanceof
	Nested classes,inner classes
Package concept, import packageName.ClassName;	import packageName*, static import, defining packages, class path
Exception concept, common exceptions	Throwing standard unchecked exceptions. Checked exceptionstry/catch/finally, throws
String, Math, Object, List, ArrayList	Sorting methods in Arrays andCollections
Wrapper classes (Integer, Double)	autoboxing