Official MSUDenver Java Subset: Language Features

This document specifies what parts of Java should be covered in CS 1050, and what parts need not be covered.

Each instructor may cover additional material as they see fit, giving priority to covering the required parts that will be assessed on the common final.

The CS Group process was to go through the similar document for the Advanced Placement Exam in CS and modify it for our local purposes.

In the following the first column for each topic contains the features of Java that may be tested in the common final, and the second column contains comments and additional features that would be good to cover but won't be tested.

---

1. The primitive types are `int`, `double`, `boolean`, and `char`.	AP leaves out `char`.
2. The five arithmetic operators, `+`, `-`, `*`, `/`, and %, are included, but only care about % with positive integer arguments.	
3. The operators `++` and `--` are included, but only in postfix form (`x++`, not `++x`), and only as independent statements—not included in expressions.	
4. The assignment operators `=`, `+=`, `-=`, `*=`, `/=`, and `%=` are included, but only as independent statements—not included in expressions except for the `int x = y = z = 1;` usage.	AP says won't use `+=` or `-=` in the update part of a `for` loop, but we allow this.
5. The relational operators `==`, `!=`, `<`, `>`, `<=`, and `>=` are included.	

---

6. The logical operators &&, `||`, and `!` are included, and students should understand that short-circuit evaluation is always done.

---

7. Would be nice to cover the ternary operator `?:`.

---

8. The primitive cast operations `(int)`, `(double)`, and `(char)` are included. Students should understand that casting from `double` to `int` truncates the decimal part.

While AP talks about rounding and truncating here, we think it would be preferable to use methods in the `Math` class.

---

9. String concatenation with `+` is included, and students should understand that non-string arguments have their `toString` method invoked.

---

10. The included escape sequences inside strings and character literals are \\, \", \n, \', and \r.

---

11. Some standard input methods are included, namely use of `Scanner`, its methods `next()`, `nextLine()`, `nextInt()`, and `nextDouble()`, with the corresponding `hasNext` methods, and construction of `Scanner` instances by `new Scanner(System.in)`, `new Scanner(new File(<fileName>))`, and `new Scanner( <some String> )`. Use of `JOptionPane` to obtain an input string is included. The methods `Integer.parseInt()` and `Double.parseDouble()` are included to allow for converting strings into these primitive values.

To read a file one `char` at a time, the closest we can come is to use `nextLine()` to get a string and then use `charAt()` to extract individual symbols. It would be good to cover using `BufferedReader` to really read a file one `char` at a time.

12. Output using `System.out.print` and `System.out.println` are included, and so are using `PrintWriter` connected to a file and the `JOptionPane` methods `showMessageDialog` and `showInputDialog`.

    Formatted output using `printf()` would be nice to cover.

13. Use of `main`, command line arguments, and use of the command line are included.

14. One and two-dimensional arrays are included, with initialization of named arrays like `int[] a = {1,2,3}`. Students should know that `.length` gives the number of array elements. The concept of a 2D array as a 1D array of 1D arrays is included.

15. The control structures `if`, `if-else`, `while`, `for`, enhanced `for`, and `do-while` are included.

    Might want to cover `switch`, `break`, and `continue`.

16. Method overloading is included, along with understanding that two methods with the same name must have different parameter lists.

17. Of course classes and `new` are included. AP committee seems confused here. These issues relating to classes belong elsewhere, not in the subset specification.

18. All classes are `public`, all instance variables are `private`, and methods must be either `public` or `private`.

19. Comments of form `//` and `/* */` are included.

    Good to also teach `javadoc` and `/** */` comments, with `@param` and `@return`.

---

20. Keyword `final` used only for `static` and block scope `final` constants.

---

21. Instance and `static` methods are included. Static methods only invoked through class name.

---

22. Instance and `static` variables are included.

---

23. The `null` reference is included.

---

24. Use of `this` is included, but only to pass reference to an object as an argument to a method call.

    Nice to cover `this( <args>)` in a constructor.

---

25. Use of `super` to invoke a superclass constructor or method is included.

---

26. Constructors that initialize all instance variables are included. Static variables should be initialized where declared.

---

27. Inheritance is not included. Interfaces are not included.

    Inheritance should be covered at a conceptual level, so students can understand how `System.out.print( x )` can automatically invoke the `toString` method in the correct class.

---

28. Students are not expected to understand what it means for a class to implement an interface.

---

29. Topic of `==` versus `.equals()` is included, and students should be able to implement an `equals` method for a class, like any other relatively simple method.

---

30. Cloning is not included.

---

31. `finalize` is not included.

---

32. Students need not understand subclass and superclass references issues.

---

33. Students should understand importing classes from packages one at a time.

Nice to also teach the `*` feature.

---

34. Nested and inner classes are not included.

---

35. Students need to understand using predefined classes that use type parameters such as `ArrayList<Whatever>`, but need not be able to implement classes or methods with such parameters.

---

36. Enumerations, annotations, and threads are not included.

---

37. Handling of exceptions by simple `try-catch` is included. Using `throws` is included.