Ultimate Java Quick Reference - CodeWithHarry

1. // [comment]

Single line comment.

2. /* [comment] */

Multi line comment.

3. public

This can be imported publically.

4. import [object].*

Imports everything in object.

5. static

Going to be shared by every [object].

6. final

Cannot be changed; common to be defined with all uppercase.

7. double

Integer with numbers that can have decimals.

8. ;

Put after every command.

9. String

Just a string of characters.

10. Private

Can only be changed by a method.

11. int

Can store numbers from 2^-31 to 2^31.

12. fields are attributes

13. boolean

Can have true or false as the value.

14. { }

These are used to start and end a function, class, etc.

15. byte

These can store from -127 - 128.

16. long

Can store numbers from 2^127 to 2^-127.

17. char

Just lets you put in one chracter.

18. double

64-bit number with decimals.

19. float

32-bit number with decimals.

20. protected

Can only be accessed by other code in the package.

21. Scanner

This lets you get user input.

22. new [object constructor]

This will let you create a new object.

23. System.in

This lets you get data from the keyboard.

24. public [class]()

This will be the constructor, you use it to create new objects.

25. super()

This will create the superclass (the class it's inheriting).

26. extends [class]

Makes the object a subclass of [object], [object] must be a superclass.

27. ++

Will increment the amount.

28. --

Will decrement the amount.

29. += [amount]

Increment by [amount]

30. -= [amount]

Decrement by [amount]

31. *= [amount]

Multiply by [amount]

32. /= [amount]

Divide by [amount]

33. System.out.println([text])

Will print something to the output console.

34. +

Can be used for concatenation. (ex. "6" + [var_here])

35. public static void main(String[] args)

This is your main function and your project will start in here.

36. System.out.print([text])

This prints stuff but there is no line break. (/n)

37. \n

Called a line break; will print a new line.

38. \t

This will print a tab.

39. if ([condition])

This will make it so if [condition] is true then it'll keep going.

40. &&

This means and.

41. !

This means not.

42. ||

This means or.

43. ==

This means equal to.

44. <

This means less than.

45. >

This means greater than.

46. >=

This means greater than or equal to.

47. [inputVarHere].hasNextLine()

This will return if there is a next line in the input.

48. this

Refer to the class that you are in.

49. [caller].next[datatype]()

This will get the [datatype] that you somehow inputted.

50. Create getters and setters

This will create the get methods and set methods for every checked variable.

51. [caller].hasNext[datatype]()

This will return if it has the correct datatype within the input.

52. overloading

If you have different parameters you can call them whatever way you want.

53. parameters

These are the inputs of your function.

54. ([datatype])[variable]

This will convert [variable] into [datatype]. Also known as casting.

55. Math.random()

Generate an extremely percise string of numbers between 0 and 1.

56. Primitives

Just the basic data types which are not objects.

57. [x].toString()

Will convert [x] into a string.

58. [number].parse[numbertype]([string])

This will parse [number] into the [numbertype] with [string].

59. ^

Return true if there is one true and one false.

60. !=

Not equal too. (NEQ)

61. ([condition]) ? [amount] : [var]

This will be like a shortcut way to an if statement.

62. switch([variable])

This will do stuff with specific cases. (e.g. switch(hi){ case 2: (do stuff)})

63. case [value]:

This will do stuff if the case is the case.

64. break

Put that when you want to leave the loop/switch; should be at end of case.

65. default [value]:

This will do stuff if none of the cases in the switch statement was made.

66. for ([number]; [condition]; [operation])

This will start at [number] and then do [operation] until [condition] is met.

67. continue

This will just go back to the enclosing loop before reaching other code.

68. while ([condition])

This will basically do something while [condition] is true.

69. void

This means no return type.

70. return

This will return something when you call it to where it was called from .

71. do { } while ([condition])

Guarantees it will execute once even if [condition] isn't met.

72. printf("%[type] stuff here bah bla", [variable here])

This will let you use [variable here] with %s being where.

73. System.out.printf([text])

Another way to print? // didn't quite get but ok then

74. [type] [returntype] [name]([parameters]) {

This is a way to create a method.

75. [type][[indexes]]

This will create an array with [indexes] amount of indexes; default infinite.

76. int[] something = new int[20];

This will just make an array of ints with 20 ints in it.

77. for ([object] [nameOfObject] : [arrayOfObject]) {

This will iterate through all of the arrayOfObject with object in use incrementing by 1 until done.

78. [object][[1]][[2]][[3]] [name] = {[value] [value] [value] \n [value] [value] [value]}

[1] is how many down in array, [2] how many accross in array, [3] how many groups

79. .length

This will get how long something is, text, amount of indexes in array, etc.

80. Arrays.copy0f([array], indexes);

This will copy the array and how many indexes into another array.

81. Arrays.toString([array])

Convert the whole array into one huge string.

82. Arrays.binarySearch([array], [object])

This will search for [object] in [array].

Kadi Sarva Vishvavidhyalaya
LDRP Institute of Technology and Research, Gandhinagar

CE/IT Department
Practical List
Subject: Object Oriented Programming with Java (CT 405 N)

Sr No	Aim of Practical	Tentative Completion Date
1	Program to implement a Class and method without parameters.	07 January 2023
2	Program to implement a Class and method with parameters and return value.	07 January 2023
3	Program to implement Stack.	21 January 2023
4	Program to implement object passing as arguments.	21 January 2023
5	Program to implement this keyword.	28 January 2023
6	Program to implement Call by value.	28 January 2023
7	Program to implement Call by reference.	04 February 2023
8	Program to implement returning objects.	04 February 2023
9	Program to implement Recursion (Example of factorial).	04 February 2023
10	Program to implement Recursion (Printing elements of an Array).	11 February 2023
11	Program to implement Access Control.	11 February 2023
12	Program to implement the use of static keyword.	18 February 2023
13	Program to implement inner classes.	18 February 2023
14	Program to implement Command Line Argument.	25 February 2023
15	Program to implement Inheritance.	25 February 2023
16	Program to implement Method Overriding.	04 March 2023
17	Program to implement super keyword.	04 March 2023
18	Program to implement final keyword.	04 March 2023
19	Program to implement Package.	11 March 2023
20	Program to implement interface.	11 March 2023
21	Program to implement partial implementation.	11 March 2023
22	Program to implement Dynamic stack implementing Interface.	11 March 2023
23	Program to implement Exception Handling.	18 March 2023
24	Program to implement user defined Exception.	18 March 2023
25	Program to implement main Thread.	25 March 2023
26	Program to create new threads.	25 March 2023
27	Program to implement multiple threads.	25 March 2023
28	Program to demonstrate synchronize keyword.	01 April 2023
29	Program to implement thread priority.	01 April 2023
30	Program to solve producer consumer problem using threads.	01 April 2023
31	Program to demonstrate deadlock, using thread.	08 April 2023
32	Program to read characters from console using I/O classes.	08 April 2023
33	Program to read lines of strings from console.	08 April 2023
34	Program to copy the contents of one file to another file.	08 April 2023
35	Program to describe the Applet skeleton.	15 April 2023
36	Program to display a moving banner in applet.	15 April 2023
37	Program to demonstrate the properties of HTML tag and getDocumentBase and getCodeBase methods.	23 April 2023
38	Program that demonstrate the mouse event handlers.	23 April 2023

Subject Coordinator: Tejasvee Gupta