1. abstract
   1. a class or method in java declared abstract. Only an abstract class or an interface may have abstract methods. Abstract methods and classes are declared without an implementation.
   2. abstract void moveTo(double deltaX, double deltaY)
2. Assert
   1. Used to verify the correctness of an invariant in the code. Used when debugging, not used in production code. They are indicative of a bug. They can be activated at run-time with the –ea option on the java command.
   2. Result = fooService.read(id);

Assert result != null;

Return result

1. Boolean
   1. A variable that returns either true or false (1 or 0 respectively)
   2. Boolean = class; bool = primitive type
   3. Boolean foo = true;
2. Break
   1. Exits the innermost switch, for, while, or do-while statement
   2. While(true){

break;

}

1. Byte
   1. A data type that is an 8-bit signed two’s complement integer.   
      Minimum of -128, maximum of 127 (inclusive). Primitive type.
   2. byte foo = 0;
2. Case
   1. Point of execution if the variable matches the case in a switch statement
   2. Swith(x)

Case: “man”:

X = “Matt”;

1. Catch
   1. Catches exceptions thrown by the java virtual machine during runtime. Usually part of a try catch block
   2. catch(Exception e){

//handle

}

1. Char
   1. A primitive type used to store individual 16-bit Unicode characters
   2. Char c = ‘c’;
2. Class
   1. A class is an individual blueprint for which objects are created
   2. Class Bike{

String color;

void getColor(){

return color;

}

}

1. Const
   1. A constant variable, it cannot be changed. Java uses final instead of const
   2. final int one = 1;
2. Continue
   1. Skips to the next iteration of the loop
   2. For (int I 0; I < 4; i++){

If(I == 0){

continue;}

}

1. Default
   1. The method used when there are conflicting methods. Or the case used in a switch statement when no other cases’ criteria are meet.
2. Double
   1. Primitive data type that is a double=precision 64-bit floating point
3. Do
   1. Used at the beginning of a do while loop. A do while loops expression is evaluated at the end of each iteration
4. Else
   1. Used in an if else statement, works as the default case if no other if/if else statement evaluates to true
5. Enum
   1. Set of possible value for a variable
6. Extends
   1. Used by children classes to declare their parent classes
7. False
   1. a statement that is false is a bool = 1;
8. Final
   1. Used by java to declare a constant variable
   2. Final FOO = 0;
9. Finally
   1. Always executes when a try block exits.
10. Float
    1. A primitive data type that is a single-precision 32-bit floating point
11. For
    1. A loop that can declare a variable, evaluate a condition, and change a variable each iteration
    2. For(int i= 0; i<2; i++)
12. Goto
    1. Bad to use, changes execution from the current line to the line indicated
    2. Goto 5;
13. If
    1. Executes code if the containing condition is true
       1. If(x == y)
14. Implements
    1. Used by a class that wants to use an interface
15. Import
    1. Used to incorporate class files in a program from a different package
    2. Import java.util.\*;
16. Instanceof
    1. Tests if an object is of a specified type
17. Int
    1. A primitive data type that is a 32-bit signed two’s complement integer
18. Interface
    1. Contains a method/group of methods with empty bodies
19. Long
    1. A primitive data type that is a 64-bit two’s complement integer
20. Native
    1. A method that will be implemented in another language, working with JNI
21. New
    1. Constructs a new instance of an object
22. Null
    1. An empty variable(contains nothing, different form 0)
23. Package
    1. The directory that contains java class files being accessed
24. Private
    1. A method or variable that may only be accessed within itself
25. Protected
    1. A variable or method that may be accessed only within itself or its children.
26. Public
    1. A variable or method that can be accessed outside of its class
27. Return
    1. Used at the end of a nonvoid method to return data to the calling line of code
    2. Return false;
28. Short
    1. A primitive data type that is a 16-bit signed two’s complement integer
29. Static
    1. Allows the variable/method to be accessed anywhere in the program
    2. Static void main()
30. Strictfp
    1. The containing method/class/interface must use consistent floating point values
    2. public strictfp class MyFPclass {

// ... contents of class here ...

}

1. Super
   1. Calls the super/parent class of the current class
2. Switch
   1. Used in place of if else blocks. Branches are evaluated by cases
3. Synchronized
   1. Used in threads to prevent thread interference and memory consistency errors
4. This
   1. Used to indicate variables and methods of the current/calling object
5. Throw
   1. Returns an exception
6. Throws
   1. A method that may return an exception
7. Transient
   1. marks a member variable not to be serialized when it is persisted to streams of bytes.
8. True
   1. A Boolean variable that is true, 1.
9. Try
   1. A block that when executes if it throws an exception, it may be handled with a catch statement
10. Void
    1. A method that has no return
    2. Public static void main(String args[]){}
11. Volatile
    1. A variable whose value will be modified by different threads
    2. Static volatile int x;
12. While
    1. A loop whose ending case is evaluated before entering the loop
    2. While(false){

}