# Quizlet

## 87 Multiple choice questions

**1%**

**1.** Boolean Value

INCORRECT

✕  !, &&, ||, ^

THE ANSWER

can be true or false

**2.** Relational Operators (Boolean)

INCORRECT

✕  using the (+) sign to combine strings

THE ANSWER

<, <=, ==, !=, >, >=

**3.** illegal identifier

CORRECT

✓  4thQtrSales

**4.** the amount of space between pixels, measured in millimeters

INCORRECT

✕ No answer given

THE ANSWER

dot pitch

**5.** Integrated development environment

INCORRECT

✕  +=, -=, **=, /= and %= (i+= 8 is i = i + 8)

THE ANSWER

an environment for developing Java programs

**6.** on a program denoted by //xxxxx or /***xxxx***/

INCORRECT

✕ No answer given

THE ANSWER

comment

**7.** assignment operator

INCORRECT

× No answer given

THE ANSWER

=

**8.** when else matches with the most recent if statement

INCORRECT

× No answer given

THE ANSWER

dangling else ambiguity

**9.** int

INCORRECT

× No answer given

THE ANSWER

name of type

**10.** assignment statement

INCORRECT

× No answer given

THE ANSWER

evaluates to the value to be assigned to a variable (=)

**11.** using the (+) sign to combine strings

INCORRECT

× No answer given

THE ANSWER

concatenate strings

**12.** identifier

INCORRECT

× No answer given

THE ANSWER

-names that refer to values or names - letters, digits, ‿, and $.
-rules for creating a name in a program

**13.** escape sequence

INCORRECT

× No answer given

THE ANSWER

\"

**14.** the values operated on by a operator

INCORRECT

× No answer given

THE ANSWER

operands

**15.** char

INCORRECT

× No answer given

THE ANSWER

name of type

**16.** Block Comment

INCORRECT

× No answer given

THE ANSWER

/*XXXXXXXXXXXXX*/

**17.** a very large int, more precise

INCORRECT

× No answer given

THE ANSWER

long type

**18.** a constant value that appears directly in a program

INCORRECT

× No answer given

THE ANSWER

literal

**19.** float

INCORRECT

× No answer given

THE ANSWER

name of a type

**20.** ++ placed before variable. increases variable by one, then uses it in the expression

INCORRECT

× No answer given

THE ANSWER

preincrement

**21.** interpreter

INCORRECT

✕  No answer given

THE ANSWER

translates source code into machine code

**22.** a class name in the system library that contains different java functions

INCORRECT

✕  No answer given

THE ANSWER

util

**23.** occurs when a program does not perform the way it was intended to

INCORRECT

✕  No answer given

THE ANSWER

logic error

**24.** casting a data type from a large range to a smaller range - Java does this automatically

INCORRECT

✕  No answer given

THE ANSWER

narrowing (of types)

**25.** do, else, and break

INCORRECT

✕  No answer given

THE ANSWER

keywords

**26.** \n

INCORRECT

✕  No answer given

THE ANSWER

escape character

**27.** using no breaks in a switch

INCORRECT

✕  No answer given

THE ANSWER

fall-through behavior

**28.** byte type

INCORRECT

× No answer given

THE ANSWER

-128 to 127

**29.** The part of a program where the variable can be referenced

INCORRECT

× No answer given

THE ANSWER

scope of a variable

**30.** If you try to store a value in a data type that cannot handle it.

INCORRECT

× No answer given

THE ANSWER

overflow

**31.** input, process, output - describes simple code

INCORRECT

× No answer given

THE ANSWER

IPO

**32.** Constant value directly in a program that stands for itself

INCORRECT

× No answer given

THE ANSWER

Literal

**33.** operator associativity

INCORRECT

× No answer given

THE ANSWER

determine the order in which operators are evaluated

**34.** Evaluates an expression based on a condition (pg 103)

INCORRECT

× No answer given

THE ANSWER

Conditional Expression ( ? :)

**35.** consists of a set of separate programs, each invoked from a command line, for developing and testing Java programs

INCORRECT

✕  No answer given

THE ANSWER

Java Development Toolkit

**36.** method that is applied to objects of Scanner

INCORRECT

✕  No answer given

THE ANSWER

nextDouble

**37.** String

INCORRECT

✕  No answer given

THE ANSWER

a type

**38.** errors that cause a program to terminate early, an impossible operation is detected

INCORRECT

✕  No answer given

THE ANSWER

runtime error

**39.** Occurs when the user inputs a value the program cannot handle

INCORRECT

✕  No answer given

THE ANSWER

input error

**40.** double type

INCORRECT

✕  No answer given

THE ANSWER

real numbers, decimal places, twice as precise as float

**41.** Assembler

INCORRECT

✕  No answer given

THE ANSWER

a device used to translate assembly-language programs into machine code

**42.** short circuit operator

INCORRECT

✕ No answer given

THE ANSWER

same as lazy operator - && or || (and, or)

**43.** an operation that converts a value of one data type into a value of another data type

INCORRECT

✕ No answer given

THE ANSWER

casting

**44.** +=, -=, **=, /= and %= (i+= 8 is i = i + 8)

INCORRECT

✕ No answer given

THE ANSWER

Augmented assignment operators

**45.** an exact number, 1, 4 or 10

INCORRECT

✕ No answer given

THE ANSWER

int type

**46.** 3.14159E1

INCORRECT

✕ No answer given

THE ANSWER

floating point/pi

**47.** casting from a small type to a larger type, this is done manually

INCORRECT

✕ No answer given

THE ANSWER

widening (of types)

**48.** imports all the classes in a package by using a * **(import java.util.**\*;)

INCORRECT

✕ No answer given

THE ANSWER

wildcard import

**49.** keyword

INCORRECT

✕  No answer given

THE ANSWER

Reserved words that have a specific meaning in java and cannot be used for variables

**50.** --

INCORRECT

✕  No answer given

THE ANSWER

decrement operator

**51.** scientific notation

INCORRECT

✕  No answer given

THE ANSWER

Floating point

**52.** preprocessor

INCORRECT

✕  No answer given

THE ANSWER

import statement

**53.** ++ placed after variable. uses original variable in expression then increases by 1

INCORRECT

✕  No answer given

THE ANSWER

postincrement

**54.** postdecrement

INCORRECT

✕  No answer given

THE ANSWER

-- placed after variable. uses original variable in expression then decreases by 1

**55.** Variable

INCORRECT

✕  No answer given

THE ANSWER

identifier

**56.** translates a Java source file into a Java bytecode file

INCORRECT

✕ No answer given

THE ANSWER

compiler

**57.** var++, + and -, casting, !, * / %, + - concaction, (See page 105)

INCORRECT

✕ No answer given

THE ANSWER

operator precedence

**58.** ++

INCORRECT

✕ No answer given

THE ANSWER

increment operator

**59.** constant

INCORRECT

✕ No answer given

THE ANSWER

a number in the program that never changes, denoted by "final"

**60.** denotes names

INCORRECT

✕ No answer given

THE ANSWER

final

**61.** represents a value stored in the computers memory

INCORRECT

✕ No answer given

THE ANSWER

variable

**62.** Binary digits

INCORRECT

✕ No answer given

THE ANSWER

Bit

**63.** data type

INCORRECT

✕ No answer given

THE ANSWER

the kind of data stored in each variable

**64.** statement terminator

INCORRECT

✕ No answer given

THE ANSWER

;

**65.** statement

INCORRECT

✕ No answer given

THE ANSWER

instructions for a high-level program

**66.** Bytecode

INCORRECT

✕ No answer given

THE ANSWER

Similar to machine instructions, but can run on any platform with a JVM

**67.** abstract is a

INCORRECT

✕ No answer given

THE ANSWER

keyword

**68.** floating-point number

INCORRECT

✕ No answer given

THE ANSWER

Numbers with a decimal point (var double)

**69.** operators

INCORRECT

✕ No answer given

THE ANSWER

+, -, *, /, %

**70.** 075

INCORRECT

✕ No answer given

THE ANSWER

octa integer

**71.** !, &&, II, ^

INCORRECT

✕ No answer given

THE ANSWER

boolean operators

**72.** A library in Java that contains predefined classes and interfaces

INCORRECT

✕ No answer given

THE ANSWER

Application Program Interface ( API)

**73.** represents a computation involving values, variables, and operators that, taking them together, evaluates to a value

INCORRECT

✕ No answer given

THE ANSWER

expression

**74.** An expression that evaluates a Boolean value to be true or false

INCORRECT

✕ No answer given

THE ANSWER

Boolean Expression

**75.** Block

INCORRECT

✕ No answer given

THE ANSWER

anything inside of a {xxxxxx}

**76.** denotes a value as a constant

INCORRECT

✕ No answer given

THE ANSWER

final keyword

**77.** -- placed before variable. decreases variable by one, then uses it in the expression

INCORRECT

✕ No answer given

THE ANSWER

predecrement

**78.** statements that let you choose actions with alternative choices

INCORRECT

✕ No answer given

THE ANSWER

selection statement

**79.** Byte

INCORRECT

✕ No answer given

THE ANSWER

8 bits to 1 byte

**80.** directive

INCORRECT

✕ No answer given

THE ANSWER

import

**81.** primitive data type

INCORRECT

✕ No answer given

THE ANSWER

int, real numbers, characters and booleans

**82.** Bool

INCORRECT

✕ No answer given

THE ANSWER

variable name

**83.** source code/program

INCORRECT

✕ No answer given

THE ANSWER

a high-level program's code

**84.** bytecode verifier

INCORRECT

✕  No answer given

THE ANSWER

checks the validity of a bytecode

**85.** conditional operator

INCORRECT

✕  No answer given

THE ANSWER

? : for if statement shorthand

**86.** Assembly Language

INCORRECT

✕  No answer given

THE ANSWER

uses a short descriptive word to represent each of the machine-language instructions

**87.** result from errors in code construction, such as misspellings, wrong punctuation, etc.

INCORRECT

✕  No answer given

THE ANSWER

syntax error