Quizlet

87 Multiple choice questions

1. Boolean Value

    A. denotes a value as a constant

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

    C. can be true or false

    D. !, &&, ||, ^

2. Relational Operators (Boolean)

    A. can be true or false

    B. <, <=, ==, !=, >, >=

    C. using the (+) sign to combine strings

    D. +, -, *, /, %

3. illegal identifier

    A. denotes names

    B. denotes a value as a constant

    C. 4thQtrSales

    D. 8 bits to 1 byte

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

    A. constant

    B. operands

    C. data type

    D. dot pitch

5. Integrated development environment

    A. an environment for developing Java programs

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

    C. using the (+) sign to combine strings

    D. A library in Java that contains predefined classes and interfaces

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

  A. comment

  B. constant

  C. casting

  D. dot pitch

7. assignment operator

  A. Bool

  B. --

  C. char

  D. =

8. when else matches with the most recent if statement

  A. primitive data type

  B. narrowing (of types)

  C. assignment statement

  D. dangling else ambiguity

9. int

  A. name of type

  B. escape character

  C. preprocessor

  D. name of a type

10. assignment statement

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

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

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

  D. statements that let you choose actions with alternative choices

11. using the (+) sign to combine strings

  A. Conditional Expression (? :)

  B. concatenate strings

  C. dangling else ambiguity

  D. conditional operator

12. identifier

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

  B. method that is applied to objects of Scanner

    -names that refer to values or names - letters, digits, ‿, and $.

  C. -rules for creating a name in a program

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

13. escape sequence

  A. keyword

  B. Bit

  C. operators

  D. \"

14. the values operated on by a operator

  A. variable

  B. operators

  C. operands

  D. compiler

15. char

  A. name of a type

  B. data type

  C. name of type

  D. directive

16. Block Comment

  A. /*XXXXXXXXXXXXX*/

  B. !, &&, ||, ^

  C. denotes a value as a constant

  D. do, else, and break

17. a very large int, more precise

  A. operands

  B. constant

  C. double type

  D. long type

18. a constant value that appears directly in a program

A. constant

B. dot pitch

C. literal

D. compiler

19. float

A. scope of a variable

B. name of type

C. directive

D. name of a type

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

A. statement

B. expression

C. logic error

D. preincrement

21. interpreter

A. translates source code into machine code

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

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

D. translates a Java source file into a Java bytecode file

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

A. literal

B. final

C. syntax error

D. util

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

A. postincrement

B. logic error

C. runtime error

D. literal

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

   A. selection statement

   B. postincrement

   C. assignment statement

   D. narrowing (of types)

25. do, else, and break

   A. compiler

   B. keywords

   C. int type

   D. nextDouble

26. \n

   A. increment operator

   B. escape character

   C. decrement operator

   D. name of type

27. using no breaks in a switch

   A. concatenate strings

   B. dangling else ambiguity

   C. short circuit operator

   D. fall-through behavior

28. byte type

   A. 3.14159E1

   B. import

   C. -128 to 127

   D. Variable

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

   A. widening (of types)

   B. assignment statement

   C. scope of a variable

   D. Boolean Expression

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

   A. operands

   B. overflow

   C. dot pitch

   D. literal

31. input, process, output - describes simple code

   A. IPO

   B. Bit

   C. final

   D. Block

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

   A. variable

   B. final

   C. Literal

   D. Bytecode

33. operator associativity

   A. when else matches with the most recent if statement

   B. statements that let you choose actions with alternative choices

   C. determine the order in which operators are evaluated

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

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

   A. concatenate strings

   B. conditional operator

   C. assignment statement

   D. Conditional Expression (? :)

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

   A. Java Development Toolkit

   B. logic error

   C. scope of a variable

   D. widening (of types)

36. method that is applied to objects of Scanner

   A. nextDouble

   B. data type

   C. runtime error

   D. overflow

37. String

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

   B. a type

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

   D. Bool

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

   A. runtime error

   B. preprocessor

   C. logic error

   D. postdecrement

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

   A. wildcard import

   B. double type

   C. nextDouble

   D. input error

40. double type

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

   B. an exact number, 1, 4 or 10

   C. real numbers, decimal places, twice as precise as float

   D. the kind of data stored in each variable

41. Assembler

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

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

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

   D. represents a value stored in the computers memory

42. short circuit operator

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

   B. when else matches with the most recent if statement

   C. the amount of space between pixels, measured in millimeters

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

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

   A. compiler

   B. identifier

   C. Bytecode

   D. casting

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

   A. primitive data type

   B. floating-point number

   C. Augmented assignment operators

   D. short circuit operator

45. an exact number, 1, 4 or 10

   A. int type

   B. data type

   C. long type

   D. statement

46. 3.14159E1

   A. floating point/pi

   B. octa integer

   C. Floating point

   D. postincrement

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

   A. operator precedence

   B. Java Development Toolkit

   C. widening (of types)

   D. primitive data type

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

   A. runtime error

   B. input error

   C. final keyword

   D. wildcard import

49. keyword

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

   B. Constant value directly in a program that stands for itself

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

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

50. --

   A. concatenate strings

   B. decrement operator

   C. statement terminator

   D. increment operator

51. scientific notation

   A. boolean operators

   B. fall-through behavior

   C. Floating point

   D. postincrement

52. preprocessor

   A. +, -, *, /, %

   B. import

   C. denotes a value as a constant

   D. import statement

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

   A. input error

   B. Block Comment

   C. compiler

   D. postincrement

54. postdecrement

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

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

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

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

55. Variable

A. operators

B. identifier

C. variable name

D. statement

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

A. compiler

B. casting

C. Literal

D. dot pitch

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

A. operator precedence

B. Conditional Expression (? :)

C. floating-point number

D. short circuit operator

58. ++

A. operator precedence

B. increment operator

C. preincrement

D. decrement operator

59. constant

A. statements that let you choose actions with alternative choices

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

C. real numbers, decimal places, twice as precise as float

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

60. denotes names

   A. Boolean Value

   B. final keyword

   C. int type

   D. final

61. represents a value stored in the computers memory

   A. data type

   B. variable

   C. expression

   D. literal

62. Binary digits

   A. IPO

   B. Bit

   C. operators

   D. final

63. data type

   A. an exact number, 1, 4 or 10

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

   C. the amount of space between pixels, measured in millimeters

   D. the kind of data stored in each variable

64. statement terminator

   A. \n

   B. ++

   C. --

   D. ;

65. statement

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

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

   C. instructions for a high-level program

   D. int, real numbers, characters and booleans

66. Bytecode

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

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

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

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

67. abstract is a

   A. final keyword

   B. keyword

   C. keywords

   D. statement

68. floating-point number

   A. when else matches with the most recent if statement

   B. Numbers with a decimal point (var double)

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

   D. same as lazy operator - && or ll (and, or)

69. operators

   A. !, &&, ll, ^

   B. +, -, *, /, %

   C. abstract is a

   D. the values operated on by a operator

70. 075

   A. byte type

   B. variable name

   C. octa integer

   D. floating point/pi

71. !, &&, ll, ^

   A. operators

   B. decrement operator

   C. bytecode verifier

   D. boolean operators

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

A. Boolean Expression

B. Application Program Interface (API)

C. selection statement

D. operator precedence

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

A. Assembler

B. postdecrement

C. expression

D. preincrement

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

A. wildcard import

B. Boolean Expression

C. Boolean Value

D. postdecrement

75. Block

A. input, process, output - describes simple code

B. anything inside of a {xxxxxx}

C. abstract is a

D. an exact number, 1, 4 or 10

76. denotes a value as a constant

A. Boolean Value

B. final keyword

C. Block Comment

D. runtime error

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

A. preincrement

B. runtime error

C. postincrement

D. predecrement

78. statements that let you choose actions with alternative choices

A. selection statement

B. Boolean Expression

C. assignment statement

D. conditional operator

79. Byte

A. Binary digits

B. abstract is a

C. +, -, *, /, %

D. 8 bits to 1 byte

80. directive

A. import

B. float

C. Variable

D. char

81. primitive data type

A. int, real numbers, characters and booleans

B. using the (+) sign to combine strings

C. real numbers, decimal places, twice as precise as float

D. statements that let you choose actions with alternative choices

82. Bool

A. directive

B. name of type

C. variable name

D. Boolean Value

83. source code/program

A. a high-level program's code

B. checks the validity of a bytecode

C. translates source code into machine code

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

84. bytecode verifier

A. when else matches with the most recent if statement

B. checks the validity of a bytecode

C. the kind of data stored in each variable

D. a constant value that appears directly in a program

85. conditional operator

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

B. ? : for if statement shorthand

C. using no breaks in a switch

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

86. Assembly Language

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

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

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

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

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

A. postincrement

B. preincrement

C. syntax error

D. util