Quizlet

NAME_____

87 Multiple choice questions

1. the values operated on by a operator

   A. overflow

   B. compiler

   C. Bytecode

   D. operands

2. primitive data type

   A. using the (+) sign to combine strings

   B. statements that let you choose actions with alternative choices

   C. int, real numbers, characters and booleans

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

3. name of a type

   A. int

   B. float

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

   D. !, &&, ||, ^

4. postdecrement

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

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

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

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

   D. -rules for creating a name in a program

5. bytecode verifier

   A. the kind of data stored in each variable

   B. using the (+) sign to combine strings

   C. checks the validity of a bytecode

   D. the values operated on by a operator

6. double type

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

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

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

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

7. 3.14159E1

A. illegal identifier

B. floating point/pi

C. Byte

D. byte type

8. literal

A. a constant value that appears directly in a program

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

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

D. method that is applied to objects of Scanner

9. boolean operators

A. do, else, and break

B. checks the validity of a bytecode

C. !, &&, ||, ^

D. scientific notation

10. statements that let you choose actions with alternative choices

A. widening (of types)

B. primitive data type

C. selection statement

D. assignment statement

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

A. scope of a variable

B. narrowing (of types)

C. widening (of types)

D. selection statement

12. +, -, *, /, %

A. logic error

B. overflow

C. dot pitch

D. operators

13. Block

A. a very large int, more precise

B. anything inside of a {xxxxxx}

C. abstract is a

D. denotes a value as a constant

14. Java Development Toolkit

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

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

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

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

15. 4thQtrSales

A. octa integer

B. byte type

C. Byte

D. illegal identifier

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

A. final

B. variable

C. Literal

D. statement

17. Assembly Language

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

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

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

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

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

A. dangling else ambiguity

B. Augmented assignment operators

C. floating-point number

D. short circuit operator

19. input error

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

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

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

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

D. -rules for creating a name in a program

20. escape character

A. ++

B. \n

C. statements that let you choose actions with alternative choices

D. Bool

21. decrement operator

A. char

B. \n

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

D. --

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

A. Application Program Interface (API)

B. Conditional Expression (? :)

C. concatenate strings

D. compiler

23. postincrement

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

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

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

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

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

A. logic error

B. compiler

C. runtime error

D. postdecrement

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

A. long type

B. Literal

C. Bytecode

D. constant

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

A. Assembler

B. Literal

C. overflow

D. variable

27. util

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

B. a high-level program's code

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

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

28. translates source code into machine code

A. predecrement

B. statement

C. interpreter

D. operator associativity

29. a type

A. identifier

B. String

C. operands

D. directive

30. name of type

A. \n

B. the kind of data stored in each variable

C. a very large int, more precise

D. char

31. conditional operator

A. statements that let you choose actions with alternative choices

B. the kind of data stored in each variable

C. using the (+) sign to combine strings

D. ? : for if statement shorthand

32. Integrated development environment

A. using the (+) sign to combine strings

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

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

D. an environment for developing Java programs

33. operator associativity

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

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

C. translates source code into machine code

D. determine the order in which operators are evaluated

34. variable name

A. char

B. represents a value stored in the computers memory

C. int

D. Bool

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

A. compiler

B. variable

C. comment

D. constant

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

A. variable

B. compiler

C. Assembler

D. keyword

37. assignment operator

A. =

B. --

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

D. using the (+) sign to combine strings

38. Boolean Expression

A. statements that let you choose actions with alternative choices

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

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

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

39. int type

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

B. instructions for a high-level program

C. an exact number, 1, 4 or 10

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

40. abstract is a

A. keyword

B. final keyword

C. operators

D. Block

41. escape sequence

A. IPO

B. Bit

C. \"

D. keywords

42. wildcard import

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

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

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

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

43. dangling else ambiguity

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

B. when else matches with the most recent if statement

C. using the (+) sign to combine strings

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

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

A. concatenate strings

B. short circuit operator

C. scope of a variable

D. dot pitch

45. floating-point number

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

B. Numbers with a decimal point (var double)

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

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

46. using no breaks in a switch

A. short circuit operator

B. final keyword

C. fall-through behavior

D. dangling else ambiguity

47. final

A. denotes names

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

C. abstract is a

D. Binary digits

48. keywords

A. /***XXXXXXXXXXXXX***/

B. !, &&, ||, ^

C. do, else, and break

D. import statement

49. method that is applied to objects of Scanner

A. Assembler

B. expression

C. nextDouble

D. statement

50. Variable

A. Boolean Value

B. identifier

C. final keyword

D. String

51. Boolean Value

A. checks the validity of a bytecode

B. /***XXXXXXXXXXXXX***/

C. can be true or false

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

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

A. runtime error

B. source code/program

C. input error

D. syntax error

53. Floating point

A. /***XXXXXXXXXXXXX***/

B. Numbers with a decimal point (var double)

C. can be true or false

D. scientific notation

54. casting

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

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

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

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

55. instructions for a high-level program

A. postdecrement

B. casting

C. predecrement

D. statement

56. import statement

A. postdecrement

B. final keyword

C. preprocessor

D. operators

57. the kind of data stored in each variable

A. data type

B. long type

C. operands

D. statement

58. comment

A. represents a value stored in the computers memory

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

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

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

59. import

A. identifier

B. directive

C. preprocessor

D. int type

60. Block Comment

A. scientific notation

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

C. denotes a value as a constant

D. /***XXXXXXXXXXXXX***/

61. using the (+) sign to combine strings

A. fall-through behavior

B. concatenate strings

C. conditional operator

D. floating-point number

62. expression

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

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

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

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

D. -rules for creating a name in a program

63. narrowing (of types)

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

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

C. Numbers with a decimal point (var double)

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

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

A. runtime error

B. wildcard import

C. postincrement

D. preincrement

65. 075

A. name of a type

B. variable name

C. octa integer

D. data type

66. IPO

A. an environment for developing Java programs

B. input, process, output - describes simple code

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

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

67. source code/program

A. a high-level program's code

B. checks the validity of a bytecode

C. determine the order in which operators are evaluated

D. ? : for if statement shorthand

68. Application Program Interface (API)

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

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

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

D. an environment for developing Java programs

69. assignment statement

A. statements that let you choose actions with alternative choices

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

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

D. when else matches with the most recent if statement

70. a very large int, more precise

A. logic error

B. nextDouble

C. long type

D. Boolean Value

71. denotes a value as a constant

A. preprocessor

B. final keyword

C. input error

D. runtime error

72. short circuit operator

A. int, real numbers, characters and booleans

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

C. when else matches with the most recent if statement

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

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

A. Conditional Expression (? :)

B. assignment statement

C. narrowing (of types)

D. operator precedence

74. ;

A. decrement operator

B. statement terminator

C. bytecode verifier

D. assignment operator

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

A. variable

B. Assembler

C. compiler

D. Bytecode

76. Bit

A. Binary digits

B. scientific notation

C. +, -, *, /, %

D. abstract is a

77. represents a value stored in the computers memory

A. variable

B. int type

C. data type

D. literal

78. dot pitch

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

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

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

D. the values operated on by a operator

79. ++

A. name of type

B. assignment operator

C. increment operator

D. decrement operator

80. identifier

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

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

B. -rules for creating a name in a program

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

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

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

A. Assembler

B. postincrement

C. predecrement

D. wildcard import

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

A. statement

B. preincrement

C. postincrement

D. expression

83. 8 bits to 1 byte

A. Byte

B. final

C. Bytecode

D. illegal identifier

84. keyword

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

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

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

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

85. -128 to 127

A. data type

B. byte type

C. illegal identifier

D. long type

86. Relational Operators (Boolean)

A. !, &&, ||, ^

B. ? : for if statement shorthand

C. +, -, *, /, %

D. <, <=, ==, !=, >, >=

87. int

A. name of a type

B. name of type

C. Boolean Value

D. data type