**87-- Multiple choice questions**

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

**-rules for creating a name in a program**

a. runtime error

b. input error

c. identifier

d. expression

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

a. floating-point number

b. assignment statement

c. widening (of types)

d. narrowing (of types)

**4. abstract is a---**

a. keyword

b. statement

c. int type

d. final keyword

**5. escape sequence**

1. final

2. keywords

3. Bit

4. \"

**6. Numbers with a decimal point (var double)**

1. floating-point number

2. concatenate strings

3. Floating point

4. widening (of types)

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

1. preincrement

2. logic error

3. identifier

4. compiler

**8. float**

1. name of type

2. name of a type

3. variable name

4. data type

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

1. dangling else ambiguity

2. selection statement

3. floating-point number

4. short circuit operator

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

1. assignment statement

2. dangling else ambiguity

3. short circuit operator

4. Conditional Expression ( ? :)

**11. /\*XXXXXXXXXXXXX\*/**

1. Block Comment

2. postincrement

3. preprocessor

4. Floating point

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

1. postdecrement

2. preincrement

3. statement

4. postincrement

**13. an environment for developing Java programs**

1. Conditional Expression ( ? :)

2. Integrated development environment

3. nextDouble

4. Augmented assignment operators

**14. instructions for a high-level program**

1. statement

2. nextDouble

3. postdecrement

4. data type

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

1. assignment statement

2. widening (of types)

3. Boolean Expression

4. assignment operator

**16. checks the validity of a bytecode**

1. Bytecode

2. bytecode verifier

3. widening (of types)

4. logic error

**17. 3.14159E1**

1. floating point/pi

2. octa integer

3. Boolean Value

4. illegal identifier

**18. ++**

1. name of type

2. escape character

3. Boolean Expression

4. increment operator

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

1. runtime error

2. predecrement

3. wildcard import

4. postdecrement

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

1. statement

2. casting

3. keyword

4. comment

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

1. Assembler

2. statement

3. int type

4. constant

**22. do, else, and break**

1. keyword

2. keywords

3. operands

4. operators

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

1. Boolean Expression

2. boolean operators

3. scope of a variable

4. dangling else ambiguity

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

1. narrowing (of types)

2. dangling else ambiguity

3. floating-point number

4. runtime error

**25. =**

1. assignment statement

2. increment operator

3. assignment operator

4. decrement operator

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

1. syntax error

2. Bytecode

3. source code/program

4. interpreter

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

1. conditional operator

2. concatenate strings

3. primitive data type

4. fall-through behavior

**28. denotes names**

1. final

2. final keyword

3. Bit

4. Block

**29. import**

1. directive

2. preprocessor

3. identifier

4. name of type

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

1. runtime error

2. logic error

3. input error

4. postincrement

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

1. operator precedence

2. wildcard import

3. floating-point number

4. input error

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

1. compiler

2. comment

3. variable

4. Assembler

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

1. assignment statement

2. operator precedence

3. selection statement

4. Application Program Interface ( API)

**34. --**

1. assignment statement

2. decrement operator

3. postdecrement

4. name of type

**35. 8 bits to 1 byte**

1. illegal identifier

2. operators

3. octa integer

4. Byte

**36. \n**

1. variable name

2. name of type

3. decrement operator

4. escape character

**37. ? : for if statement shorthand**

1. Relational Operators (Boolean)

2. selection statement

3. Conditional Expression ( ? :)

4. conditional operator

**38. a type**

1. directive

2. String

3. operands

4. identifier

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

1. overflow

2. logic error

3. scope of a variable

4. dot pitch

**40. int**

1. int type

2. name of type

3. data type

4. final keyword

**41. denotes a value as a constant**

1. final keyword

2. final

3. input error

4. preprocessor

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

1. preincrement

2. keywords

3. keyword

4. overflow

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

1. Block Comment

2. postdecrement

3. statement

4. postincrement

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

1. operands

2. variable

3. data type

4. statement

**45. int, real numbers, characters and booleans**

1. dangling else ambiguity

2. selection statement

3. short circuit operator

4. primitive data type

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

1. constant

2. Literal

3. Bytecode

4. int type

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

1. literal

2. util

3. Assembler

4. casting

**48. !, &&, ||, ^**

1. Floating point

2. bytecode verifier

3. Boolean Expression

4. boolean operators

**49. 075**

1. floating point/pi

2. name of type

3. identifier

4. octa integer

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

1. Boolean Expression

2. primitive data type

3. operator precedence

4. selection statement

**51. on a program denoted by //xxxxx or /\*xxxx\*/**

1. comment

2. casting

3. double type

4. constant

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

1. Literal

2. Assembler

3. statement

4. nextDouble

**53. determine the order in which operators are evaluated**

1. operator associativity

2. Conditional Expression ( ? :)

3. interpreter

4. assignment statement

**54. scientific notation**

1. postincrement

2. Floating point

3. boolean operators

4. final keyword

**55. <, <=, ==, !=, >, >=**

1. Application Program Interface ( API)

2. operators

3. Floating point

4. Relational Operators (Boolean)

**56. a high-level program's code**

1. bytecode verifier

2. source code/program

3. concatenate strings

4. interpreter

**57. can be true or false**

1. double type

2. Boolean Value

3. bytecode verifier

4. long type

**58. ;**

1. widening (of types)

2. increment operator

3. statement terminator

4. assignment statement

**59. import statement**

1. preprocessor

2. preincrement

3. expression

4. operators

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

1. overflow

2. compiler

3. Assembler

4. dot pitch

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

1. identifier

2. Java Development Toolkit

3. postdecrement

4. expression

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

1. concatenate strings

2. scope of a variable

3. operator precedence

4. Conditional Expression ( ? :)

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

1. Conditional Expression ( ? :)

2. assignment statement

3. scope of a variable

4. compiler

**64. char**

1. data type

2. name of type

3. runtime error

4. int type

**65. translates source code into machine code**

1. predecrement

2. interpreter

3. util

4. logic error

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

1. variable

2. Assembler

3. predecrement

4. overflow

**67. +, -, \*, /, %**

1. operands

2. preprocessor

3. long type

4. operators

**68. the kind of data stored in each variable**

1. data type

2. nextDouble

3. statement

4. dot pitch

**69. Variable**

1. preprocessor

2. identifier

3. directive

4. String

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

1. variable

2. comment

3. input error

4. literal

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

1. IPO

2. variable

3. input error

4. comment

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

1. Relational Operators (Boolean)

2. Integrated development environment

3. Application Program Interface ( API)

4. widening (of types)

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

1. dot pitch

2. operands

3. variable

4. overflow

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

1. Boolean Expression

2. dangling else ambiguity

3. Augmented assignment operators

4. assignment statement

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

1. dot pitch

2. selection statement

3. bytecode verifier

4. scope of a variable

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

1. postincrement

2. predecrement

3. logic error

4. preincrement

**77. real numbers, decimal places, twice as precise as float**

1. long type

2. dot pitch

3. double type

4. int type

**78. Binary digits**

1. directive

2. Block

3. Literal

4. Bit

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

1. preincrement

2. expression

3. runtime error

4. identifier

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

1. wildcard import

2. overflow

3. input error

4. double type

**81. Bool**

1. variable name

2. preincrement

3. name of a type

4. predecrement

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

1. operands

2. compiler

3. Literal

4. overflow

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

1. constant

2. compiler

3. input error

4. double type

**84. -128 to 127**

1. long type

2. byte type

3. illegal identifier

4. int type

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

1. Assembler

2. runtime error

3. wildcard import

4. Assembly Language

**86. anything inside of a {xxxxxx}**

1. Block

2. IPO

3. keyword

4. final

**87. 4thQtrSales**

1. identifier

2. floating point/pi

3. illegal identifier

4. preprocessor

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

1. int type

2. double type

3. constant

4. long type