

School of Computing and Information Technologies

PROGCON - CHAPTER 2

20

CLASS NUMBER: # 14

Data Type

Acoting point Identitier

enc constant

garion Notation

Integer

iran operator

ragic Humber

rurers values

cermoras due body

gover

Hierarchy chart

portable

Declaration

para piction

Prorrt

NAME: HEFALAK, QUAN PAUL O.

SECTION: ACIAL

DATE: 11/8/19

PART 1: Identify the following.

4. A classification that describes what values can be assigned, how the variable is stored, and what types of operations can be performed with the variable.

A diagram that illustrates modules' relationships to each other.

3. A list of every variable name used in a program, along with its type, size, and description. functional contribute to the degree to which all the module statements contribute to the same task.

5. A message-that is displayed on a monitor to ask the user for a response and perhaps explain how that response should be formatted.

6. A module that can more easily be reused in multiple programs.

A number with decimal places.

8 A program component's name.

9 A specific numeric value.

10. A statement that provides a data type and an identifier for a variable.

11. A variable-naming convention in which a variable's data type or other information is stored as

part of its name.

12. A whole number. 13. An operator that requires two operands—one on each side.

14. An unnamed constant whose purpose is not immediately apparent.

Shakeert 15. Assigns a value from the right of an assignment operator to the variable or constant on the left of the assignment operator.

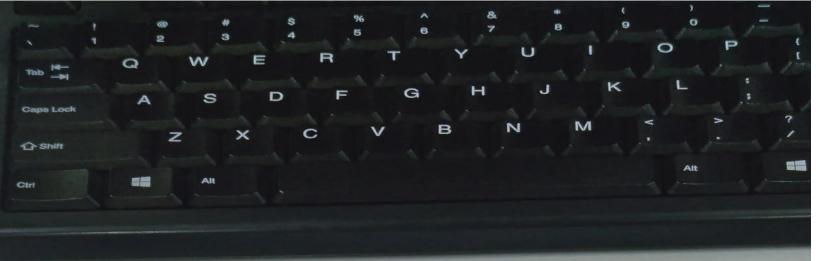
16. Can contain alphabetic characters, numbers, and punctuation.

17. Constitute the limited word set that is reserved in a language.

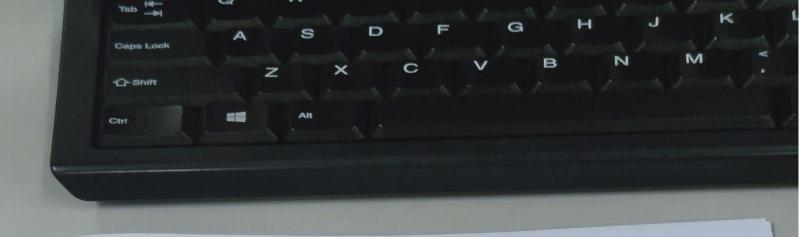
18. Contains all the statements in the module.

19. Contains information that expands on what appears in another flowchart symbol; it is most ation Symbol often represented by a three-sided box that is connected to the step it references by a dashed

20. Contains meaningful data and module names that describe the program's purpose.



- 21. Describe operators that evaluate the expression to the right first. right to tell osco cicklyity
- 23. Describes operators that evaluate the expression to the left first. left to -right accordantly
- 24. Describes the extra resources a task requires.
- 25. Describes the rules of precedence. order of
- 26. Describes the state of data that is visible. in scope
- 27. Describes the unknown value stored in an unassigned variable.
- 28. Describes variables that are declared within the module that uses them.
- 29 Describes variables that are known to an entire program. 30. Dictate the order in which operations in the same statement are carried out. Mee of precedence
- 31 Documentation that is outside a coded program. External powentation
- 32 Documentation within a coded program. Interval oscarentation
- 34. Hold the steps you take at the end of the program to finish the application. era- of job +acks 33. Floating point numbers. Real humbers
- 35. Include steps your must perform at the beginning of a program to get ready for the rest of the
- program. Howevering tostes 36. Include the steps that are repeated for each set of input data. detail loop tocks
- 37. Includes the module identifier and possibly other necessary identifying information. was the header
- 38. Is another name for the camel casing naming convention. lower coses cosing
- 39 is sometimes used as the name for the style that uses dashes to separate parts of a name. Keebole coce
- 40 Marks the end of the module and identifies the point at which control returns to the program or module that called the module. wodure return
- 41. One that can hold digits, have mathematical operations performed on it, and usually can hold a decimal point and a sign indicating positive or negative. nureal
- 42. Runs from start to stop and calls other modules. wan program
- 43. Similar to a variable, except that its value cannot change after the first assignment. Noved constant
- 44. Small program units that you can use together to make a program; programmers also refer to modules as subroutines, procedures, functions, or methods. wodules
- 45. The act of assigning its first value, often at the same time the variable is created. Intializing the variable
- 46. The act of containing a task's instructions in a module. Frequency
- 47. The act of reducing a large program into more manageable modules. Functional pecomposition
- 48. The act of repeating input back to a user either in a subsequent prompt or in output. Echoing Input
- 49. The equal sign; it is used to assign a value to the variable or constant on its left. occionred operator
- 50. The feature of modular programs that allows individual modules to be used in a variety of applications. rev savility



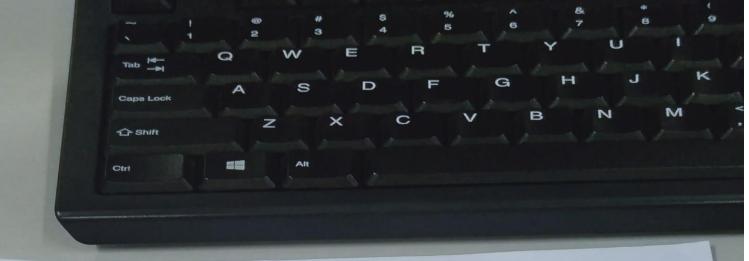
- 51. The feature of modular programs that assures you a module has been tested and proven to
- 52. The format for naming variables in which the initial letter is lowercase, multiple-word variable function correctly. reliability names are run together, and each new word within the variable name begins with an uppercase
- 53 The format for naming variables in which the initial letter is uppercase, multiple-word variable names are run together, and each new word within the variable name begins with an uppercase
- The logic that appears in a program's main module; it calls other modules. waisher to give
- 55. The memory address identifier to the left of an assignment operator. Lyalve
- 56. The process of breaking down a program into modules. Hodulonzation 57. The process of paying attention to important properties while ignoring nonessential details.
- 58 To use the module's name to invoke it, causing it to execute.
- 59. Where global variables are declared. Program lovel 60. Written explanations that are not part of the program logic but that serve as documentation for those reading the program. program com

Choose from the following

- 1. Abstraction _2. Alphanumeric values
- Annotation symbol
- Assignment operator - 4.
- Assignment statement
- 6. Binary operator
- 7. Call a module
- _ 8. Camel casing
- /9. Data dictionary
- /10. Data type
- , 11. Declaration
- / 12. Detail loop tasks
- _ 13. Echoing input
- 14. Encapsulation
- / 15. End-of-job tasks
- , 16. External documentation , 37. Main program
- 17. Floating-point
- ≥18. Functional cohesion
- 19. Functional decomposition 40. Module body
- /20. Garbage
- /21. Global

- / 22. Hierarchy chart
- / 23. Housekeeping tasks
- /24. Hungarian notation
- /25. Identifier
- /26. In scope
- ∠27. Initializing the variable
- _28. Integer
- 29. Internal documentation
- 30. Kebob case
- ✓ 31. Keywords
- / 32. Left-to-right associativity
- /33. Local
- 34. Lower camel casing
- / 35. Lvalue
- > 36. Magic number
- / 38. Mainline logic
- 39. Modularization
- _41. Module header
- / 42. Module return statement

- /43. Modules
- _44. Named constant
- _ 45. Numeric
- / 46. Numeric constant (literal numeric constant)
- 47. Numeric variable
- -48. Order of operations
- /49. Overhead
- _ 50. Pascal casing
- -51. Portable
 - 52. Program comments
- > 53. Program level
- 54. Prompt
- /55. Real numbers
- _56. Reliability
- 57. Reusability
- ✓ 58. Right-associativity and right-to-left associativity
- > 59. Rules of precedence
 - , 60. Self-documenting





School of Computing and Information Technologies

PROGCON - CHAPTER 2

checked by: Hannarexy &

CLASS NUMBER: # 14

NAME: HERALAN, ALLAM PAU O.

SECTION: DATE:

PART 2: Identify whether each variable name is valid, and if not explain why.

3 pt a) Age total

b) age_* Invalia torcitors allowed

c) +age invalid

5 pts: 60 special characters cilowed

B) Tage Invated on by constructed with digits and letters

to pts' spars would be indicated with an inderscore

PROGCON - CHO2

2nd TERM, AY2019-2020

MS. JEN