

School of Computing and Information Technologies

## 1by: Deither Sonia

## **PROGCON - CHAPTER 2**

CLASS NUMBER: # 31

SECTION: BSTM-14RO 191

NAME: Torres, Icaiah

DATE: 8/11/19

## PART 1: Identify the following.

data type . 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.

hierarchy chart 2. A diagram that illustrates modules' relationships to each other.

ot data dictionary 3. A list of every variable name used in a program, along with its type, size, and description.

hi functional collection 4. A measure of 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.

Plaking point 7. A number with decimal places.
identifier 8. A program component's name.

numeric constant 9. Aspecific numeric value.

Cleclaration 20. A statement that provides a data type and an identifier for a variable.

hongorian nolalion. 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.

binary operator 13. An operator that requires two operands—one on each side.

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

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

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

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

Leywords

18. Contains all the statements in the module. module body

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

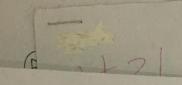
self documenting 20. Contains meaningful data and module names that describe the program's purpose.

if to left associative 21. Describe operators that evaluate the expression to the right first. 22. Describes data that consists of numbers. eft to right association Describes operators that evaluate the expression to the left first. overhead 24. Describes the extra resources a task requires. Describes the rules of precedence. Rules of ... Precedence in scope 26. Describes the state of data that is visible. 27. Describes the unknown value stored in an unassigned variable. garbage 28. Describes variables that are declared within the module that uses them. 16ca7 29. Describes variables that are known to an entire program. 910001 30. Dictate the order in which operations in the same statement are carried out. Rules of Precedence of aton the external documentation that is outside a coded program. internal documentation within a coded program. real numbers 38. Floating-point numbers. prend of - job tasts 34. Hold the steps you take at the end of the program to finish the application. 35. Include steps you must perform at the beginning of a program to get ready for the rest of the it housekeeping cletails loop fasks program. tasks module header 37. Includes the module identifier and possibly other necessary identifying information. lower camel caring 38. Is another name for the camel casing naming convention. Keboo care 35. Is sometimes used as the name for the style that uses dashes to separate parts of a name. module refund. Marks the end of the module and identifies the point at which control returns to the program or module that called the module. statement 41. One that can hold digits, have mathematical operations performed on it, and usually can hold a numeric variable decimal point and a sign indicating positive or negative. marn program 42. Buns from start to stop and calls other modules. pained constant 43. Similar to a variable, except that its value cannot change after the first assignment. 44. Small program units that you can use together to make a program; programmers also refer to modules mødules as subroutines, procedures, functions, or methods. in Fig 12/119 9 45. The act of assigning its first value, often at the same time the variable is created. variable 46. The act of containing a task's instructions in a module. functional olecond The act of reducing a large program into more manageable modules. echoing input 48. The act of repeating input back to a user either in a subsequent prompt or in output. 49. The equal sign; it is used to assign a value to the variable or constant on its left. 50. The feature of modular programs that allows individual modules to be used in a variety of reusability applications.



den

ea edi er





and informatic stability 51. The feature of modular programs that assures you a module has been tested and proven to N - CHAT function correctly. Kebob case (2) The format for naming variables in which the initial letter is lowercase, multiple-word variable names are run together, and each new word within the variable name begins with an uppercase pascal case 3 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 can mainline logic 54. The logic that appears in a program's main module; it calls other modules. Lualue 58. The memory address identifier to the left of an assignment operator. am, Modularization 56. The process of breaking down a program into modules. odul. Abstraction 87. The process of paying attention to important properties while ignoring nonessential details. ask Call a module 58. To use the module's name to invoke it, causing it to execute.

10 program level 59. Where global variables are declared. 60. Written explanations that are not part of the program logic but that serve as documentation for program Comment S those reading the program.

## Choose from the following

choose from the following		
1. Abstraction 2. Alphanumeric values	22. Hierarchy chart 23. Housekeeping tasks	43. Modules 44. Named constant
3. Annotation symbol	24. Hungarian notation	45. Numeric
A. Assignment operator	25. Identifier	46. Numeric constant (literal
5. Assignment statement	26. In scope	numeric constant)
8. Binary operator	27. Initializing the variable	47. Numeric variable
7. Call a module	28. Integer	48. Order of operations
8. Camel casing	29. Internal documentation	49. Overhead
9. Data dictionary	30. Kebob case	50. Pascal casing
10. Data type	31. Keywords	51. Portable
11. Declaration	32. Left-to-right associativity	52. Program comments
12. Detail loop tasks	33. Local	53. Program level
13. Echoing input	34. Lower camel casing	54. Prompt
14. Encapsulation	35. Lvalue	55. Real numbers
15. End-of-job tasks	36. Magic number	56. Reliability
16. External documentation	37. Main program	57. Reusability
17. Floating-point	38. Mainline logic	58. Right-associativity and
18. Functional cohesion	39. Modularization	right-to-left associativity
19. Functional decomposition	40. Module body	59. Rules of precedence
20. Garbage	41. Module header	60. Self-documenting
21. Global	42. Module return statement	