



School of Computing and Information Technologies

PROGCON - CHAPTER 2

CLASS NUMBER: 19

SECTION: AC192

NAME: RIEGO, MA. MONIQUE ISABEL S.

DATE: _____

PART 1: Identify the following.

DATA TYPE

1. 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.

DATA DICTIONARY

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

FUNCTIONAL COHESION

4. A measure of the degree to which all the module statements contribute to the same task.

PROMPT

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.

PORTABLE MODULE

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

FLOATING POINT

7. A number with decimal places.

IDENTIFIER

8. A program component's name.

NUMERIC CONSTANT

9. A specific numeric value.

DECLARATION

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

HUNGARIAN
NOMENCLATURE

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

INTEGER

12. A whole number.

BINARY OPERATOR

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

MAGIC NUMBER

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

ASSIGNMENT STATEMENT

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

ALPHANUMERIC
STRING VARIABLE

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

KEY WORD

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

MODULE BODY

18. Contains all the statements in the module.

ANNOTATION SYMBOL

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

SELF DOCUMENTING

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

- IN RIGHT ASSOCIATIVITY AND NUMERIC CWD*
21. Describe operators that evaluate the expression to the right first.
 22. Describes data that consists of numbers.
 23. Describes operators that evaluate the expression to the left first.
 24. Describes the extra resources a task requires.
 25. Describes the rules of precedence.

- IN SCOPE, GARBAGE LOCAL GLOBAL ORDER OF PRECEDENCE*
26. Describes the state of data that is visible.
 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.

EXTERNAL DOCUMENTATION

31. Documentation that is outside a coded program.

INTERNAL DOCUMENTATION

32. Documentation within a coded program.

REAL NUMBERS

33. Floating-point numbers.

END-OF-JOB-TASKS

34. Hold the steps you take at the end of the program to finish the application.

*DETAIL LOOP TASKS
HOUSEKEEPING TASKS*

35. Include steps you must perform at the beginning of a program to get ready for the rest of the program.

DETAIL LOOP TASKS

36. Include the steps that are repeated for each set of input data.

MODULE HEADER

37. Includes the module identifier and possibly other necessary identifying information.

PASCAL CASE

38. Is another name for the camel casing naming convention.

KEBAB CASE

39. Is sometimes used as the name for the style that uses dashes to separate parts of a name.

MODULE RETURN STATEMENT

40. Marks the end of the module and identifies the point at which control returns to the program or module that called the module.

NUMERIC VARIABLE

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.

MAIN PROGRAM

42. Runs from start to stop and calls other modules.

NAMED CONSTANT

43. Similar to a variable, except that its value cannot change after the first assignment.

MODULES

44. Small program units that you can use together to make a program; programmers also refer to modules as subroutines, procedures, functions, or methods.

INITIALIZING THE VARIABLE

45. The act of assigning its first value, often at the same time the variable is created.

ENCAPSULATION

46. The act of containing a task's instructions in a module.

FUNCTIONAL DECOMPOSITION

47. 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.

ASSIGNMENT OPERATOR

49. The equal sign; it is used to assign a value to the variable or constant on its left.

REUSABILITY

50. The feature of modular programs that allows individual modules to be used in a variety of applications.



School of Computing and Information Technologies

PROGCON - CHAPTER 2

CLASS NUMBER:

SECTION:

NAME: _____

DATE: _____

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

a) Age VALID

b) age_* INVALID
- asterisk are not allowed

c) +age INVALID
- the variable name are only allowed to start with the letter of an alphabet & underscore and dotsign.

d) age_ VALID

e) _age VALID

f) Age VALID

g) 1age INVALID
- the variable starts with a numerical value and it should not be allowed

h) Age 1 INVALID
- spaces are not allowed

MS. JEN

2nd TERM, AY2019-2020