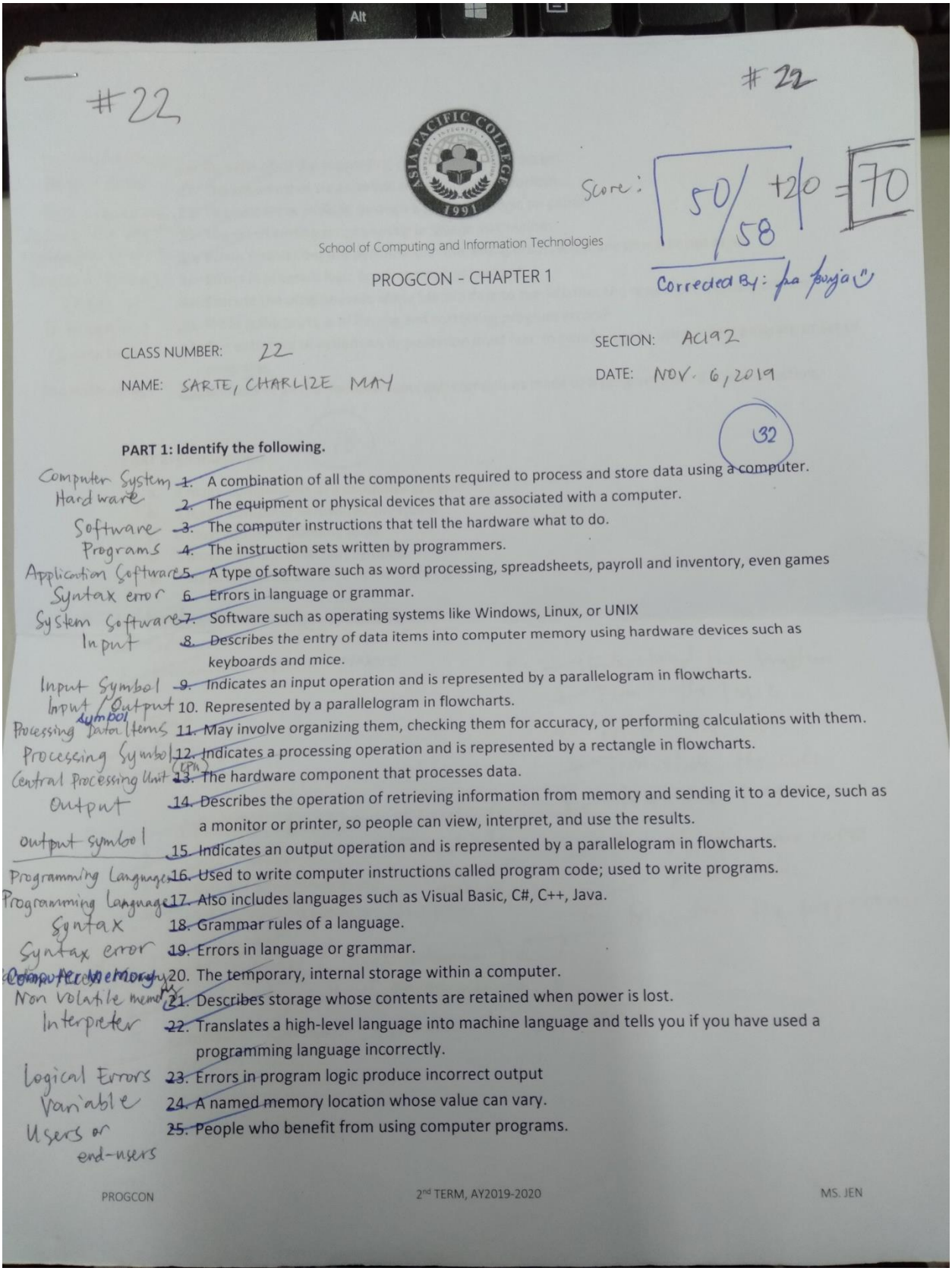
Activity # 5

Chapter 1 - Written Activity Score: 50 + 20 = **70**



#22 #22

School of Computing and Information Technologies

PROGCON - CHAPTER 1

Score: 50/58 +20 = 70

Corrected By: ma'am Sunja

CLASS NUMBER: 22 SECTION: AC192

NAME: SARTE, CHARLIZE MAY DATE: NOV. 6, 2019

32

**PART 1: Identify the following.**

1. Computer System — A combination of all the components required to process and store data using a computer.
2. Hardware — The equipment or physical devices that are associated with a computer.
3. Software — The computer instructions that tell the hardware what to do.
4. Programs — The instruction sets written by programmers.
5. Application Software — A type of software such as word processing, spreadsheets, payroll and inventory, even games
6. Syntax error — Errors in language or grammar.
7. System Software — Software such as operating systems like Windows, Linux, or UNIX
8. Input — Describes the entry of data items into computer memory using hardware devices such as keyboards and mice.
9. Input Symbol — Indicates an input operation and is represented by a parallelogram in flowcharts.
10. Input / Output — Represented by a parallelogram in flowcharts.
11. Processing Data Items — May involve organizing them, checking them for accuracy, or performing calculations with them.
12. Processing Symbol — Indicates a processing operation and is represented by a rectangle in flowcharts.
13. Central Processing Unit (CPU) — The hardware component that processes data.
14. Output — Describes the operation of retrieving information from memory and sending it to a device, such as a monitor or printer, so people can view, interpret, and use the results.
15. output symbol — Indicates an output operation and is represented by a parallelogram in flowcharts.
16. Programming Language — Used to write computer instructions called program code; used to write programs.
17. Programming Language — Also includes languages such as Visual Basic, C#, C++, Java.
18. Syntax — Grammar rules of a language.
19. Syntax error — Errors in language or grammar.
20. Computer memory — The temporary, internal storage within a computer.
21. Non volatile memory — Describes storage whose contents are retained when power is lost.
22. Interpreter — Translates a high-level language into machine language and tells you if you have used a programming language incorrectly.
23. Logical Errors — Errors in program logic produce incorrect output
24. Variable — A named memory location whose value can vary.
25. Users or end-users — People who benefit from using computer programs.

Documentation 26. Consists of all the supporting paperwork for a program.
Algorithm 27. The sequence of steps necessary to solve any problem.
Desk-checking 28. The process of walking through a program's logic on paper.
Coding the program 29. The act of writing programming language instructions.
Logical/Syntax Errors 30. When instructions are performed in the wrong order, too many times, or not at all.
Logical Errors 31. Errors in program logic produce incorrect output
Test 32. Execute the program with some sample data to see whether the results are logically correct
Debugging 33. What is the process of finding and correcting program errors?
Conversion 34. The entire set of actions an organization must take to switch over to using a new program or set of programs
Maintenance 35. Consists of all the improvements and corrections made to a program after it is in production.

18

**PART 2: Enumeration**

a. 3 major components of a computer system?
b. 3 major computer hardware operations.
c. 4 most common planning tools.
d. 3 most common flowchart symbols.
e. 7 steps on a program development life cycle.

a. 1. Hardware
Software Application software
Firmware System Software

b. 1. Input
2. Processing
3. Output

c. 1. Affinity Diagram
2. Tree Diagram
3. Matrix Diagram
4. Interrelationship Diagram
Flowcharts
Pseudocode
IPO charts
HIPO charts

d. 1. Input/Output Symbol (Parallelogram)
2. Processing Symbol (Rectangle)
3. Output Symbol (Diamond)
Decision

e. 1. Understand the Problem
2. Plan the logic
3. Write the code
4. Translate the code
5. Test the program
6. Put the program into production
7. Maintain the program