Short note – OOC

01. Steps of building an OO programing

    I. Problem to solve
    II. Identifying objects that are needed
    III. Identifying classes through abstraction
    IV. Create objects from classes
    V. Assemble objects to create the solution

Properties = attributes = data members

Behavior = functions = methods = member functions

02. 05 basic features of OO programming
    - Abstraction
    - Inheritance
    - Polymorphism
    - Encapsulation
    - Information hiding

03. What is **information hiding**?
    - Hide implementation details or certain information which are internal to the class using private key word. (protecting data from an unauthorized accesses)
    - Example: Jewellery shop

04. What is **encapsulation**?
    - binds together the data and functions that manipulate the data (writing code into a single unit), and this is a sub process of data hiding.
    - Example: TV and remote

*The main difference between **data hiding** and **encapsulation** is that data hiding focus more on data security and encapsulation focuses more on hiding the complexity of the system.

05. What is **abstraction**?
    - Removing some characteristics and provide the essential things for the outside to reduce the complexity of the program.
    - Example: How can cat views for a grandma and a veterinary surgeon.

06. What is **inheritance**?
    - Inheritance enables a new class to reuse the state and behavior of old class. The new class will be able to behave as the old class as well as can behave on its own.
    - Example: Parent and child

07. What is **polymorphism**?
    - Polymorphism enables one common interface for many implementations, and for objects to act differently under different circumstances using overriding and overloading methods.

08. Ways of representing requirements for object-oriented analysis.
    - Use case diagram
    - Use case scenario
    - User stories

09. Ways of discovering classes
    - Noun/verb analysis
    - CRC method

10. Steps for noun/verb analysis

    I. Read the paragraph carefully and identify nouns and noun phrases in there.
        - common nouns -> classes (Eg: Animal, Person)
        - proper nouns -> objects (Eg: Dog, Sirisena)
    II. Check for rules and reject nouns according to them.

11. Rules for rejecting nouns
    - Redundant
    - An event or an operation
    - Outside scope of system
    - Meta – language
    - An attribute

12. Types of analysis classes
    - Entity class -> **general classes**
    - Boundary class -> **the classes which is interact with users (forms, UI etc.)**
    - Control class -> **the class which is controlling the particular use case.**

13. Relationships between classes
    - Dependency -> **uses**
    - Association -> **has a**
    - Aggregation -> **part of (weak)**
    - Composition -> **part of (strong)**
    - Inheritance -> **is a**

14. Advantages of using OOP to develop large programs
    - Reusability through inheritance
    - Flexibility through polymorphism
    - Effective problem solving

15. Structure vs class
    - Structure is not secure and cannot hide implementation details from the end users.
    - Class is more secure and can hide implementation details.