



Introduction to Programming

Distinction Task 8.3: Food Hunter

Principles question and answer sheet.

Write up to one page (maximum 500 words) addressing the following question:

During this course we have covered some design principles for programming. These include:

- Coupling
- Cohesion
- Information hiding
- Modular design

Among others.

Compare and contrast how at least two (2) of the structured programming principles we have covered relate to theory and practice of object oriented programming. i.e what are the similarities and/or differences between structured programming and object oriented programming in relation to at least 2 design principles,

Answer:

In this answer sheet, I will give out the key differences and similarities between Structured Programming and Object-Oriented Programming (OOP)

First, we need to define these key concepts to have a clear insight into their application and how the two are in contrast.

Structured Programming is to divide the program into a set of functions or modules. This term is also known as Modular Programming. Each of the functions has an assigned task that is usually executed through the main procedure. When Structured Programming is implemented, programmers will have an easier time debugging, clearly examining the code, and making modifications with ease. 2 of the most popular structured programming languages that can be mentioned are **C** and **Pascal**. In Structured Programming, the user can also create their unique functions that can be called on by the main function. After the sub-function is performed, the program normally returns to the main function to execute other tasks. The use of global variables will be involved as it can be accessed by all different functions.

Object-Oriented Programming allows programmers to use objects to portray real-life scenarios. An object can be understood as an entity that has various states and behaviors. States will represent the attributes and the data of an object, and behaviors perform services on

requests. Class is essential to creating an object, acting as a blueprint in the creation of different objects with the same attributes. OOP focuses on the user interface and makes it easier to develop different programs.

Key differences:

Criteria	Structured Programming	Object-Oriented Programming
Definition	Divides the code into modules or functions.	Based on objects which can behave with the user through an interface.
Main Focus	Divide the program into sub-functions.	Represent a program through a set of objects that contains attributes and data.
Modification	Difficult to modify.	Easier to modify.
Communication	The main program calls out different functions.	Objects communicate with each other by passing messages.
Access Specifiers	None.	Private, public, and protected.
Security	Data is not secured.	Data is secured
Code Reusability	Difficult to reuse code.	Easy to reuse code.

In week 8, we learned about coupling and cohesion.

Coupling means how task implementations interact with each other to make modules as independent as possible, i.e. low coupling.

- ➔ Pro: Coupling eliminates unnecessary relationships & the number of necessary relationships.
 - Fewer cross-section problems.
 - Easy to maintain, changes to 1 module would not require changes to another.

Cohesion means how related task implementations are grouped together and creating modules whose elements are strongly related to one another. However, it is to note that high cohesion can lead to low coupling.

These designs are a part of Object-Oriented Programming (OOP), which helps programmers in the task of improving their coding skills and other skills related to architecture design. The purpose of Design phase in the Software Development Life Cycle is to produce a solution to a problem given in the SRS (Software Requirement Specification) document. The output of the design phase is Software Design Document (SDD).

End of Task.