Task NO 1.

Procedural programming and Object-Oriented Programming (OOP) are two different paradigms in programming.

Procedural programming is a linear approach to programming where the code is organized around procedures or functions, which are a series of instructions that operate on data. The focus is on defining the sequence of steps that need to be executed to accomplish a task. Procedural programming is often used in smaller projects where there is a clear set of steps to be executed.

On the other hand, Object-Oriented Programming (OOP) is a programming paradigm that is based on the concept of objects, which are instances of classes that encapsulate data and behavior. The focus is on modeling the real world and creating objects that interact with each other to accomplish tasks. OOP is often used in larger projects where the complexity of the code requires a more modular approach.

In terms of learning, OOP offers several benefits over procedural programming. For example:

Code reusability:

OOP allows for the creation of reusable code through the use of classes and objects, which can be instantiated multiple times in different parts of the code.

Encapsulation: OOP promotes encapsulation, which means that data and behavior are hidden inside objects, making the code more secure and easier to maintain.

Inheritance: OOP allows for the creation of new classes that inherit attributes and methods from existing classes, which can save time and reduce the amount of code needed.

Polymorphism: OOP allows for the use of polymorphism, which means that objects can take on different forms or behaviors depending on the context in which they are used.

In conclusion, while procedural programming and OOP are two different programming paradigms, OOP offers several benefits in terms of code organization, reusability, and security. Learning OOP can be useful for larger projects where there is a need for modular and scalable code.