3-1 Journal: The benefits of Modular programming

Marie-Chantal Foster

Cybersecurity, Southern New Hampshire University

CS-500-10018-M01 Introduction to Programming

Dr. Shadha Tabatabai

November 30, 2025

Modular programming

Modular programming is a software design technique that separates a computer program into independent, interchangeable modules. This paper discusses real-world scenarios, how they are implemented, how modular techniques improve code, the benefits, and the challenges (*Modular approach in programming, 2025)*.

A real-world scenario where it is highly effective is the development of an e-commerce processing system like Amazon. A system like this requires several distinct operations, including handling customer information, managing the shopping cart, processing payments, and logging transaction data. This will allow each distinct operation to be developed, tested, and maintained in isolation, helping to scale when needed and debug (*Python & SQL Bible, 2023)*. As large as Amazon is, it is tough to write code in one module, so I break it down into separate programs. In Diagram 1 example, the main program is called *main.py*, with subprograms *customerinfo.py, payments.py,* and *shoppingcart.py*. Each program will have its own file, which will then be imported into the main module shown in Diagram 3. In this example, I have the tab open for *customerinfo.py* (Diagram 2). I wrote the code and defined my function, then in *main.py* (Diagram 3), I will use *import* to include the *customerinfo.py* module. A module is a file containing code that is included in a program, and an extensive program can be split into reusable, separate files (Wassberg, 2020).

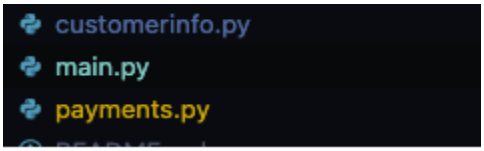

Diagram 1

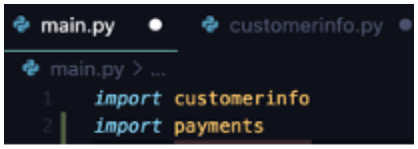

Diagram 3

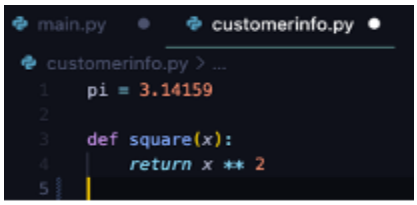

Diagram 2

When writing a program, and in this scenario, the foundation relies on "define, call, and return" functions to pass information smoothly between independent modules (Wassberg, 2020). I will use Diagram 2 as an example. I "*define"* the function as def square(x):, this will calculate the square of a number 'x'. It accepts one argument and returns the result as return x ** 2. The "*return*" statement is used to exit the function and return a result. In the *main.py* file, I would "*call"* the functions defined in the *import customerinfo* statement, which imports the module to perform the specific task. Diagram 4 shows the calculations of *customerinfo.py*. The "*call*" is executing the functions by using the function's name, followed by parentheses containing the necessary arguments (Wassberg, 2020).

Diagram 4

As Amazon's business grows, so will the code, and to maintain that organization, modular programming helps keep it in order by breaking it into smaller, self-contained, manageable modules. As shown in Diagram 4, when I work with the *customerinfo.py* module, I can separate any concerns I might have when navigating through the code to understand and manage a specific task. Adding to an organization's tasks would improve reusability, ease maintenance, facilitate debugging, and support team collaboration. For example, if I were on a team working on this extensive program, each programmer could tackle their own module concurrently, which can speed up the project without conflicts (Chiaramonte, 2024). However, the implications could be a code that is unmanageable and too large to understand. It could be tangled code, making it difficult for someone to understand, which makes changes more difficult and affects the entire system. Moreover, poor reusability would make it hard to extract and reuse the code to create other projects (*Python & SQL Bible, 2023*).

Modular programming offers key benefits for programmers. *Readability*- The benefit is that it is easy on the eyes and does not feel overwhelming, making it easier to understand what I am working on and navigate the code (Chiaramonte, 2024). *Reusability*-When the functions or classes within the module can be reused in different parts of the application to use in a newly created module. For example, I could use all or some of the *customerinfo.py* module in a new module called *ordermanagement.py*. *Testing and debugging*- Regarding debugging the code, instead of looking at thousands of code from the main module to find the mistake, I can focus on that one specific module to see it. Same as testing the code, simplifying it to minimize the risk of introducing errors across the entire system. *Maintenance*- If I continued to test, update, and debug regularly using a Kanban, it would be a powerful tool for tracking my module tasks as well as others (Chiaramonte, 2024).

Let us review a few potential challenges and how to mitigate them. *Testing*- when changes are frequent, it can be time-consuming. To mitigate, one would use a combination of individual unit tests, implementing a multi-layered testing strategy. *Skill and Planning requirements*- when inexperienced teams are under pressure, they may split tasks inefficiently, which could lead to an unmanageable system over time. To mitigate, take the time to train programmers on the processes to be followed to ensure consistent execution. *Integration and interface complexity*: if modules must interact but the boundaries are not clearly defined, then it can lead to rewrites during integration. To mitigate, use a transparent, detailed wireframe or system design diagram before getting started; if planning the backend interfaces, the flowchart diagrams will show how the components connect (SailingByte, 2023).

Modular programming is a core concept for better managing complex systems and making coding easier to build, test, modify, and maintain.

References

Chiaramonte, M. (2024, September 19). *Developing modular software: Top strategies and best practices.* Vfunction. https://vfunction.com/blog/modular-software/

Modular approach in programming. (2025, July 11). Geeksforgeeks. https://www.geeksforgeeks.org/software-engineering/modular-approach-in-programming/

Python and SQL bible: From beginner to world expert (1st ed.). (2023). Cuantum Technologies LLC.

SailingByte. (2023). *What is modular programming? All the pros and cons of the method*. SailingByte.https://sailingbyte.com/blog/what-is-modular-programming-all-the-pros-and-cons-of-the-method/

Wassberg, J. (2020). *Computer programming for absolute beginners*. Packt Publishing Ltd.