ASSIGNMENT 2.1
Supplementary Activities to Creating Modules and Packages
PSMDS1RC103
BASAL, RAFFY

1. Why do built-in functions exist?

- Built-in functions in Python simplify coding by providing optimized, ready-to-use solutions for common tasks, reducing the need for custom implementations.
- Functions offer advantages like modularity, reusability, and better organization compared to sequential code, which can be simpler but prone to duplication and harder to manage as it grows.
- A function is a single unit of reusable code, while a module is a file containing multiple functions and other definitions.
- A package, on the other hand, is a directory of related modules organized to manage code more effectively.

2. What are the advantages/disadvantages of placing code inside functions vs. sequential codes?

Functions help organize code by breaking complex problems into smaller, reusable parts, making the code easier to understand, maintain, and test. They prevent code duplication and encapsulate logic, but calling functions adds some overhead and can make the code harder to follow if not documented properly or if there are too many layers.

Sequential code is simpler and more direct for small tasks, with no overhead from function calls. However, it can lead to repetitive code and becomes difficult to manage and debug as the program grows, making it less scalable for larger projects.

In conclusion, using functions instead of sequential code enhances code organization, reusability, and maintainability. Functions allow you to encapsulate repetitive tasks into reusable units, reducing code duplication and making it easier to manage and update. Sequential code, while straightforward for simple tasks, can become cumbersome and less flexible as complexity grows. Functions provide a cleaner, more modular approach that simplifies maintenance and improves overall code efficiency.

3. What is the difference between a function and a module?

A **function** is a single, reusable piece of code that performs a specific task, taking inputs and providing outputs. It helps break down complex problems into smaller, manageable parts. In contrast, a **module** is a file that contains multiple functions, as well as other elements like classes and variables.

Modules help organize related pieces of code into a single file, making it easier to manage and reuse code across different parts of a program. Essentially, **functions** are the building blocks of code, while

modules serve as containers that group these functions and other definitions into a cohesive unit, aiding in better code organization and modularity.

In summary, a function is a single unit of reusable code, while a module is a file that groups multiple functions, classes, and variables together.

4. Discuss the difference between a module and a package.

A **module** is a single Python file that contains code such as functions, classes, and variables. It serves as a container for related pieces of code, allowing you to organize and reuse code efficiently. Modules help in structuring code into manageable components, making it easier to maintain and understand.

A **package** is a directory that contains multiple Python modules and an `__init__.py` file, which can be empty or include initialization code for the package. Packages help organize related modules into a hierarchical structure, making it easier to manage larger projects with many modules.

To summarize, a module is a single file that contains functions, classes, and variables, while a package is a directory that contains multiple modules and an `__init__.py` file, serving as a container for organizing related modules.