**Week 4 – Notes from the Lecture**

**Advantages of Functions**

1. Code Reusability: Functions allow you to write code once and reuse it whenever needed. This reduces redundancy and saves time.

2. Modularity: By breaking a large program into smaller, self-contained functions, it becomes easier to understand, debug, and maintain. Each function can handle a specific task.

3. Code Organization: Functions help to organize code into logical sections, improving readability and making complex programs easier to navigate.

4. Abstraction: Functions provide a way to encapsulate code logic. Users of the function don’t need to understand the internal workings; they just need to know what the function does.

5. Simplified Testing and Debugging: Functions allow you to test small, isolated parts of the code individually, making it easier to find and fix errors.

6. Improves Maintainability: Functions allow easier updates. If you need to modify the logic, you only need to update the function, and all places where it’s used will benefit from the change.

**Syntax for Defining a Function in Python**

To define a function in Python, you use the `def` keyword, followed by the function name, parameters (if any), and a colon. The function body is indented and may contain one or more statements. Optionally, you can return a value using the `return` statement.

Example:

def function\_name(parameters):

# Function body

# Optional return statement

return value

Example:

def greet(name):

print(f"Hello, {name}!")

greet("Alice") # Output: Hello, Alice!

**What is Needed to Use Functions Stored in a Module?**

To use functions stored in a module, you need to import the module into your script using the `import` statement. Once imported, you can access the module’s functions using dot notation.

Steps:

1. Import the module:

- Use `import module\_name` to import the entire module.

- Or use `from module\_name import function\_name` to import specific functions.

2. Call the function:

- If the whole module is imported, you call the function with `module\_name.function\_name()`.

- If specific functions are imported, you call the function directly as `function\_name()`.

Example:

import math # Import the entire math module

print(math.sqrt(16)) # Use the sqrt() function from the math module

from math import pi # Import only the pi constant

print(pi) # Use pi directly

In this example, the `math` module is imported, and then its `sqrt()` function and `pi` constant are used.