

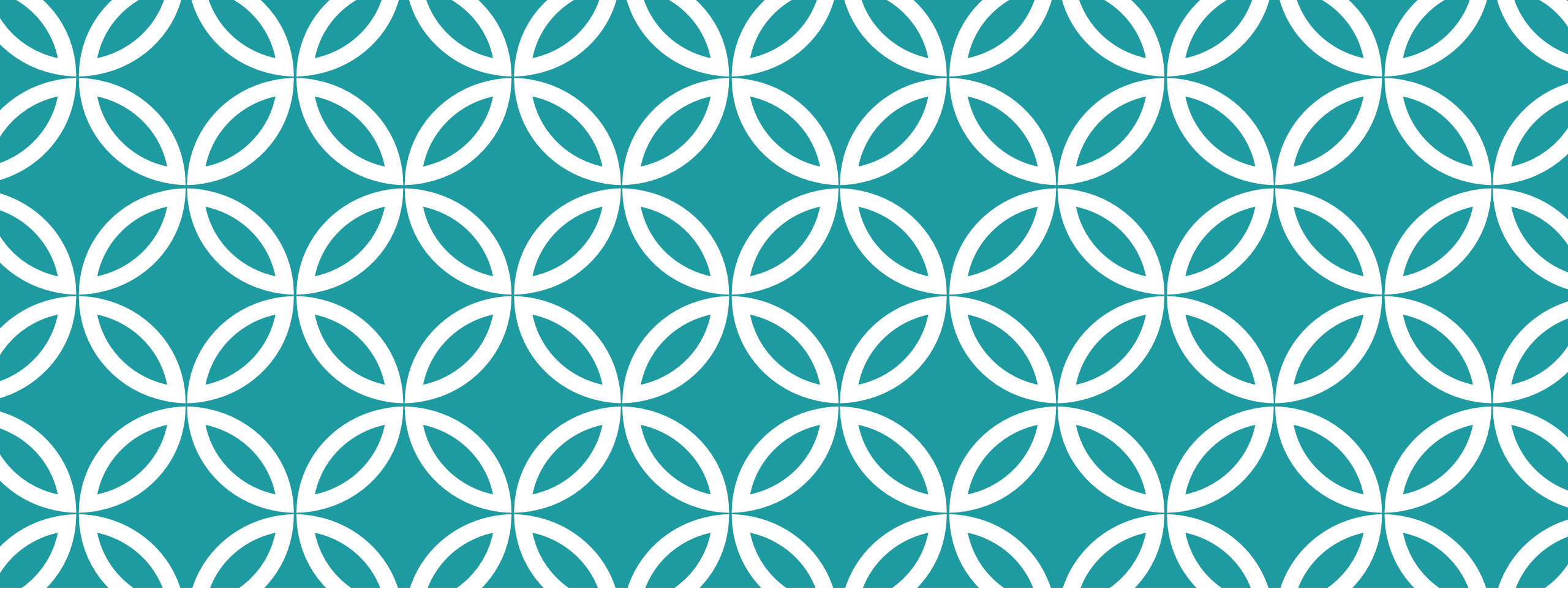


ADVANCED-PROGRAMMING

DSE – 2125

DR. MANJUNATH V HEGDE





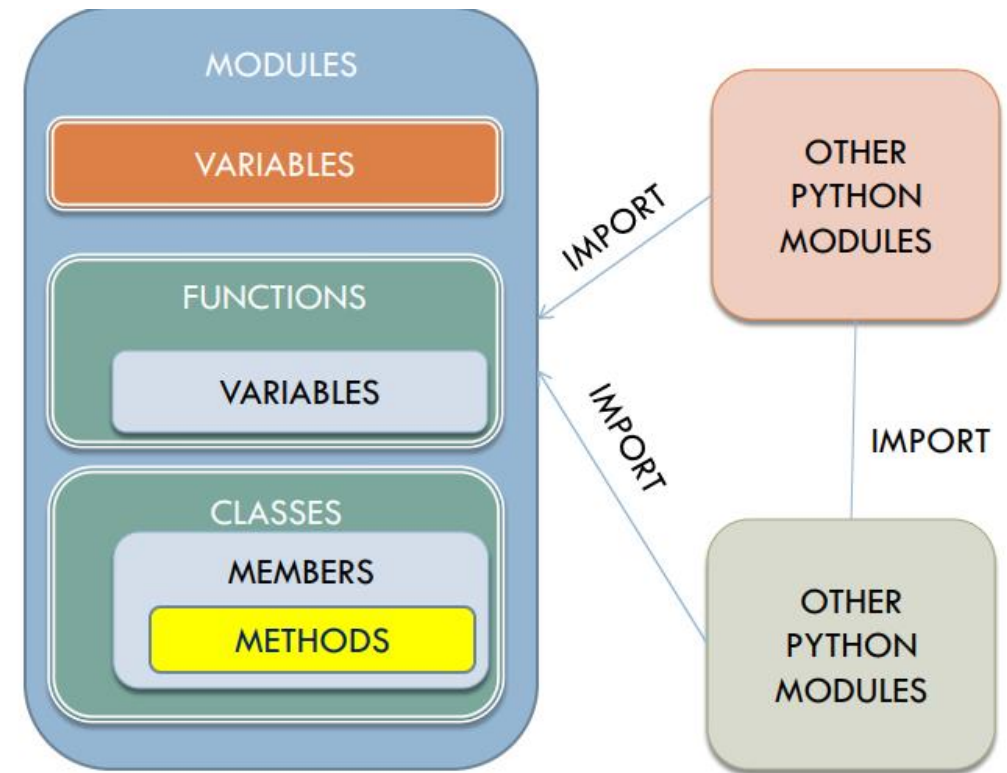
PYTHON MODULE

INTRODUCTION

Modules are used to categorize Python code into smaller parts.

A module is simply a Python file where statements, classes, objects, functions, constants and variables are defined.

Grouping similar code into a single file makes it easy to access.



ADVANTAGES OF PYTHON MODULES

Putting code into modules is useful because of the ability to import the module functionality.

A module can be used in some other Python code. Hence, it provides the facility of code reusability.

A module allows us to logically organize our Python code.

Grouping related code into a module makes the code easier to understand and use.

Similar types of attributes can be categorized and placed in a single module

MODULE NAMES

A module name is the file name with the .py extension.

When we have a file named empty.py, empty is the module name.

The `__name__` is a variable that holds the name of the module being referenced.

IMPORTING PYTHON MODULES

The most common way to create a module is to define a separate file containing the code we want to group separately from the rest of the application.

Modules also make it easier to reuse the same code in more than one program. If we have written a set of functions that is needed in several different programs, we can place those functions in a module.

We can use any Python source file as a module by executing an **import** statement.

IMPORTING PYTHON MODULES

“import” statement can be used to import a module. It is the simplest and most common way to use modules in our code. It gives access to all attributes (like variables, constants, etc.) and methods present in the module.

- `import <module_name>`

In case of accessing specific modules, it can be modified as:

- `import module1[, module2 [, ...moduleN]`

To access particular methods from any module, the statement shall be:

- `<module_name>.<function_name>`

FROM STATEMENT

Python's from statement lets us import specific attributes or individual objects from a module into the current (calling) module or namespace.

- *from <module_name> import <function_name(s)>*

Or

- *from <module_name> import function_name1 [...function_name2[...function_nameN]]*

To use/access/invoke a function, we will specify the module name and name of the function separated by a dot (.). This format is also known as dot notation.

<module_name>.<function_name>

IMPORT *

`import *` statement can be used to import all names from a module into the current calling (namespace) module.

- *from <module_name> import**