



**Laboratory Manual
for
SC1003
Introduction to Computational Thinking and
Programming**

**Practical Exercise #5:
Data Abstraction**

**COLLEGE OF COMPUTING AND DATA SCIENCE
NANYANG TECHNOLOGICAL UNIVERSITY**

Ex. #5 – Data Abstraction

Learning Objectives

Understand and implement data structure in Python. Focused on Strings, Tuples, Lists, and Dictionaries in programming using the context of a Battleships game.

Equipment and accessories required

PC/notebook with python compiler (eg python IDLE)

1. Data Structure

A data structure is an organization of data in such a way to allow more efficient manipulation of the data by a program. For instance, you have been using integer and **string** in your programming exercises, which are basic (known as primitive in Python) data structure that contain single value of a data, consisting of a sequence of numbers and characters respectively. With data arranged in certain structure, you can then apply various operations to these data structure; such as arithmetic operations for integers and string manipulation (e.g. `isdigit()` function).

In this exercise, you will learn how to use composite data structures that store a collection of values (instead of single value) to simplify your program organization. They are **Tuple**, **List** and **Dictionary** (which are known as non-primitive data structure in Python).

Continue working on the code developed in Lab 4 and add the following exercises:

In this exercise, the code was enhanced to demonstrate the use of Python's **String**, **Tuple**, **List**, and **Dictionary** data structures within the context of a Battleships game. The main changes include:

1. **User Input Processing:**

- The user is prompted to enter the starting positions of the ships (e.g., Carrier and Submarine) in the format "row,col".
- The input string is split into a list of strings, which is then converted to a tuple of integers using the `map()` function. This tuple represents the starting coordinates of each ship.

2. **Dictionary Usage:**

- A dictionary is used to store details about each ship, including its name, length, and starting coordinates. This dictionary allows easy access to ship properties and organizes ship data efficiently.

3. **Ship Placement on the Board:**

- A 10x10 game board (a list of lists) is initialized with all cells set to 0 (empty).
- A function `place_ship()` is defined to place ships on the board based on their starting position and length. The ships are placed horizontally using a for loop, and their positions are marked on the board.

If you are new to Python programming, copy the following hint code to your IDE, e.g., IDLE, follow the TODO task lists and sample output as follows to complete the exercises.

```
board_size = 10

# Get the start position for the Carrier
print("Please enter start_position of Carrier in the following
format (row,col). E.g. 6,4")
```

Ex. #5 – Data Abstraction

```
user_input_c = input("Enter coordinates: ")

# TODO : 1. Split the string at the comma and convert to a
tuple of integers
# Add you code of TODO 1 here

# Get the start position for the Submarine
print("Please enter start_position of Submarine in the
following format (row,col). E.g. 6,4")
user_input_s = input("Enter coordinates: ")

# TODO : 2. Split the string at the comma and convert to a
tuple of integers
# Add you code of TODO 2 here

# TODO : 3. Dictionary to store ship details (name, length,
and starting coordinates)
# Add you code of TODO 3 here

#code from lab4
# Initialize a board_size x board_size game board with all
cells set to 0 (empty)

def place_ship(ship_name):
# TODO : 4.Function to place ships on the board based on their
start position and length
# Add you code of TODO 4 here

# Place the ships on the board
place_ship("Carrier")
place_ship("Submarine")

#code from lab4
# While loop to repeatedly ask for valid attack coordinates

#code from lab4
# For loop to iterate through each row and column of the board
```

Sample output:

```
Please enter start_position of Carrier in the following format (row,col). E.g. 6,4
Enter coordinates: 4,1
Please enter start_position of Submarine in the following format (row,col). E.g. 6,4
Enter coordinates: 5,2
Enter attack row (0-9): 6
Enter attack column (0-9): 1
Coordinates (6, 1) are valid: True
0000000000
0000000000
0000000000
0000000000
0000000000
0SSSSS0000
00SSS00000
0100000000
0000000000
```

Ex. #5 – Data Abstraction

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0