



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

**Practical Exercise #3:
Basic Python Programming -
Conditional Statements**

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

Ex. #3 – Basic Python Programming

Learning Objectives

Understand and implement conditional statements in programming using the context of a Battleships game.

Equipment and accessories required

PC/notebook with python compiler (eg python IDLE)

1. Conditional Execution

For a program to be useful and practical, it is essential that the program has the ability to check conditions and alter its behaviour accordingly. The simplest way to perform this conditional execution is to make use of the **if** statement, which has the general form of

```
if <boolean expression>:  
    statement  
^^^^  
    :
```

Note that the body of the compound statement that get executed when the condition (i.e. the boolean expression) is true must be indented by the same amount, and it is recommended to use **4 spaces** for indenting in Python.

In practice, it is more common to use the **if-else** and **if-elif-else** statements. (Refers to LAMS video for detail and examples on how they can be used.)

Exercises:

1. **Basic Conditional Statements:**
 - Write a program to check if the entered coordinates are within the valid range.
2. **Conditional with Logical Operators:**
 - Write a program to validate user input for coordinates and orientation.
3. **Nested Conditionals:**
 - Write a program to validate the placement of a ship (e.g., check for out-of-bounds placement).

Deliverables:

- Program files demonstrating the use of conditional statements in the context of the Battleships game.

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.

```
# For beginners, all code in a single file without functions  
board_size = 10  
valid_orientations = ['horizontal', 'vertical']
```

```
# Exercise 1: Basic Conditional Statements  
x, y = 5, 8
```

```
# TODO : 1. Check if coordinates are within the valid range
```

Ex. #3 – Basic Python Programming

```
# Add you code of TODO 1 here
```

```
#One more example
```

```
x, y = 10, -1
```

```
# TODO : 2. Check if coordinates are within the valid range
```

```
# Copy your code of TODO 1 here to test on x, y = 10, -1
```

```
# Exercise 2: Conditional with Logical Operators
```

```
x, y, orientation = 4, 6, 'horizontal'
```

```
# TODO : 3. Validate user input for coordinates and orientation
```

```
# Add you code of TODO 3 here
```

```
#One more example
```

```
x, y, orientation = 11, 3, 'diagonal'
```

```
# TODO : 4. Validate user input for coordinates and orientation
```

```
# Copy your code of TODO 3 here to test on x, y, orientation = 11, 3, 'diagonal'
```

```
# Exercise 3: Nested Conditionals
```

```
x, y, ship_length, orientation = 3, 5, 4, 'horizontal'
```

```
# TODO : 5. Validate the placement of a ship
```

```
# Add you code of TODO 5 here
```

```
#One more example
```

```
x, y, ship_length, orientation = 7, 7, 4, 'vertical'
```

```
# TODO : 6. Validate the placement of a ship
```

```
# Copy your code of TODO 5 here to test on x, y, ship_length, orientation = 7, 7, 4, 'vertical'
```

Sample output:

Coordinates (5, 8) are valid: True

Coordinates (10, -1) are valid: False

Input (4, 6, horizontal) is valid: True

Input (11, 3, diagonal) is valid: False

Placement (3, 5, 4, horizontal) is valid: True

Placement (7, 7, 4, vertical) is valid: False