Battleship

Battleship is a guessing game which is played on ruled grids. The player is required to guess the exact positions of the fleet of ships. The locations of the fleets are concealed from the player.

```
     0   1   2   3   4   5   6   7   8   9
   +---+---+---+---+---+---+---+---+---+---+
0  |   |   | S |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
1  |   |   | X |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
2  |   |   |   | L |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
3  | D | D |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
4  |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
5  |   |   |   |   |   |   |   |   |   | X |
   +---+---+---+---+---+---+---+---+---+---+
6  |   |   |   |   |   |   |   |   |   | X |
   +---+---+---+---+---+---+---+---+---+---+
7  |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
8  |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
9  |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
```

This diagram illustrates the empty board for player to guess the location of the fleet.

```
     0   1   2   3   4   5   6   7   8   9
   +---+---+---+---+---+---+---+---+---+---+
0  | S | S | S |   | S | S | S | H | H | H |
   +---+---+---+---+---+---+---+---+---+---+
1  |   |   |   |   |   |   | C |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
2  |   |   |   | L |   |   | C |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
3  | D | D |   | L |   |   | C |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+
4  |   | H |   | L |   |   | C |   | B |   |
   +---+---+---+---+---+---+---+---+---+---+
5  |   | H |   |   |   |   | C | T | B |   |
   +---+---+---+---+---+---+---+---+---+---+
6  |   | H |   | S |   |   |   | T | B |   |
   +---+---+---+---+---+---+---+---+---+---+
7  |   |   |   | S |   |   |   |   | B |   |
   +---+---+---+---+---+---+---+---+---+---+
8  |   | D | D | S |   | B | B | B | B |   |
   +---+---+---+---+---+---+---+---+---+---+
9  |   |   |   |   |   |   |   | D | D |   |
   +---+---+---+---+---+---+---+---+---+---+
```

This diagram illustrates the hidden formation of the fleet.

- If player guessed a position correctly, the abbreviation letter of the ship (such as `"S"` or `"D"`) will be reviewed.
- Otherwise, a `"X"` will be displayed on the board.

You are tasked to create a text-based interactive “Battleship” game.

Task 4.1
Implement the following `get_mapping()` function according to its specification given.

Function	**Specification**

`get_mapping(): dict`	This procedure read and processed the file `"ship_class_mapping.txt"`.
	
	It returns a dictionary in the following format:
	`{'Carrier': ('C', 5),`
	`'Battleship': ('B', 4),`
	`'Heavy Cruiser': ('H', 3),`
	`'Light Cruiser': ('L', 3),`
	`'Submarine': ('S', 3),`
	`'Destroyer': ('D', 2),`
	`'Torpedo Boat': ('T', 2)}`

Task 4.2
Implement `Ship` class according to the UML class diagram given. Functions `get_abbr()` and `get_size()` should return the ship's abbreviation letter and size based on the dictionary you have created in task 4.1.

Ship
- class: str
+ Ship(class: str) + get_class(): str + get_abbr(): str + get_size(): int

Task 4.3
Implement `Board` class according to the UML class diagram and attributes/methods specifications given.

Board
- board: list - guess_board: list

+ Board()
+ add_ship(ship: Ship)
+ display_board(guess_mode: bool)
+ generate_new_game()
+ guess(row: int, col: int): bool
+ power_guess(row: int, col: int): bool
+ check_win(): bool

Attributes/Methods	**Specification**
`board: list`	`board` is a 2-dimensional `list` storing the formation of the hidden fleet.
`guess_board: list`	`guess_board` is a 2-dimensional `list` storing information of player guesses; and display either "`X`" for no ship found or display abbreviation of the ship if a ship is found.
`add_ship(ship: Ship)`	`add_ship()` takes in a `Ship` object, and randomly allocate the `ship` into the `board`. A `ship` can either be placed horizontally or vertically and should not have any part being placed beyond the boarder.
`display_board(guess_mode: bool)`	`display_board()` aims to display the `board` to the user based on the `guess_mode`.

	You may refer to `sample_board.txt` to understand better about the required format. If `guess_mode` is `True`, the content of the `guess_board` will be displayed; otherwise, display the actual content of the `board`.

`generate_new_game()`	`generate_new_game()` runs through all ship classes, randomly generate a game with **1 to 3 ships per class** and add them onto the `board`.
	
	After the generation is complete, a message in the following format should be displayed:
	
	`New Game Generated.`
	
	`The following ships have entered the battlefield:`
	`Carrier (Size 5): 2`
	`Battleship (Size 4): 1`
	`Heavy Cruiser (Size 3): 3`
	`Light Cruiser (Size 3): 2`
	`Submarine (Size 3): 3`
	`Destroyer (Size 2): 3`
	`Torpedo Boat (Size 2): 1`

`guess(row: int, col: int): bool`	`guess()` takes in a pair of `row`, `col` value and check if there any `ship` can be found at this position. If the position has already been guessed before, it's treated as an invalid guess, return `False`. If no `ship` found, mark an "`X`" to the `guess_board` and return `True`. If a `ship` is found, mark its corresponding abbreviation letter to the `guess_board` and return `True`.
`power_guess(row: int, col: int): bool`	`power_guess()` is similar to `guess()`, but if a `ship` is found, **all** positions occupied by this `ship` will be reviewed automatically and be marked using its abbreviation letter on the `guess_board`.
`check_win(): bool`	`check_win()` checks if all ships have been identified. If so, return `True`; otherwise return `False`.

Task 4.4
Write a menu which has the following options. Validation of the user input is needed.

```
Choose an option below:
1) Normal Guess
2) Power Guess
3) Show Answer
4) Restart Game
5) Exit
```

The descriptions for the options can be found below.

Option	Descriptions
`Normal Guess /` `Power Guess`	Ask user to perform a normal or power guess. User should be prompted to enter the row and col number. If winning condition is met, display message and automatically restart a new game.
`Show Answer`	Review the hidden formation of the fleet to the user.
`Restart Game`	Reset the board and restart the game with a newly generated game.
`Exit`	Exit program.