Mini Project -1 (Number Guessing Game)
(Language PYTHON)

- Build a number Guessing Game in which the user selects a range.
- Assume the user selected a range from lower bound of range to Upper bound of range where both lower boundary and upper boundary are integers.
- So a random number in that range is selected by the system where the user needs to guess the number in minimum number of guesses.

Algorithm for Number Guessing Game

1. **Start of the Game:**
    - Print a welcome message and game instructions.
    - Initialize necessary variables (X, Y, secret_number, attempts, guessed).
2. **Input Validation Loop:**
    - Prompt the user to enter the lower bound (X) and upper bound (Y) of the number range.
    - Use a while loop with a try-except block to ensure:
        - The user inputs are valid integers (int() function).
        - Y is greater than X to form a valid range.
    - If input is invalid (ValueError), prompt user to enter integers again.
3. **Generate Secret Number:**
    - Use random.randint(X, Y) to generate a random integer (secret_number) within the specified range [X, Y].
    - Print a message indicating the range of numbers the secret number is within.
4. **Game Loop:**
    - Use a while loop with condition not guessed to continue until the player guesses the correct number.
    - Inside the loop:
        - Prompt the user to enter their guess (guess).
        - Use try-except to handle non-integer inputs and prompt user to enter an integer.
        - Compare guess with secret_number:
            - If guess is less than secret_number, print "Too low! Try a higher number."
            - If guess is greater than secret_number, print "Too high! Try a lower number."
            - If guess equals secret_number, set guessed to True, print a congratulatory message, and display the number of attempts (attempts).

5. **Play Again Prompt:**

    - After guessing correctly (guessed is True), prompt the user if they want to play again (play_again).
    - Convert play_again to lowercase for consistency (play_again.lower()).
    - If play_again is "yes", call number_guessing_game() again to start a new game.
    - If play_again is "no" or any other input, print a farewell message and end the game.

6. **End of Game:**

    - Print "Thank you for playing!" to indicate the end of the game session.

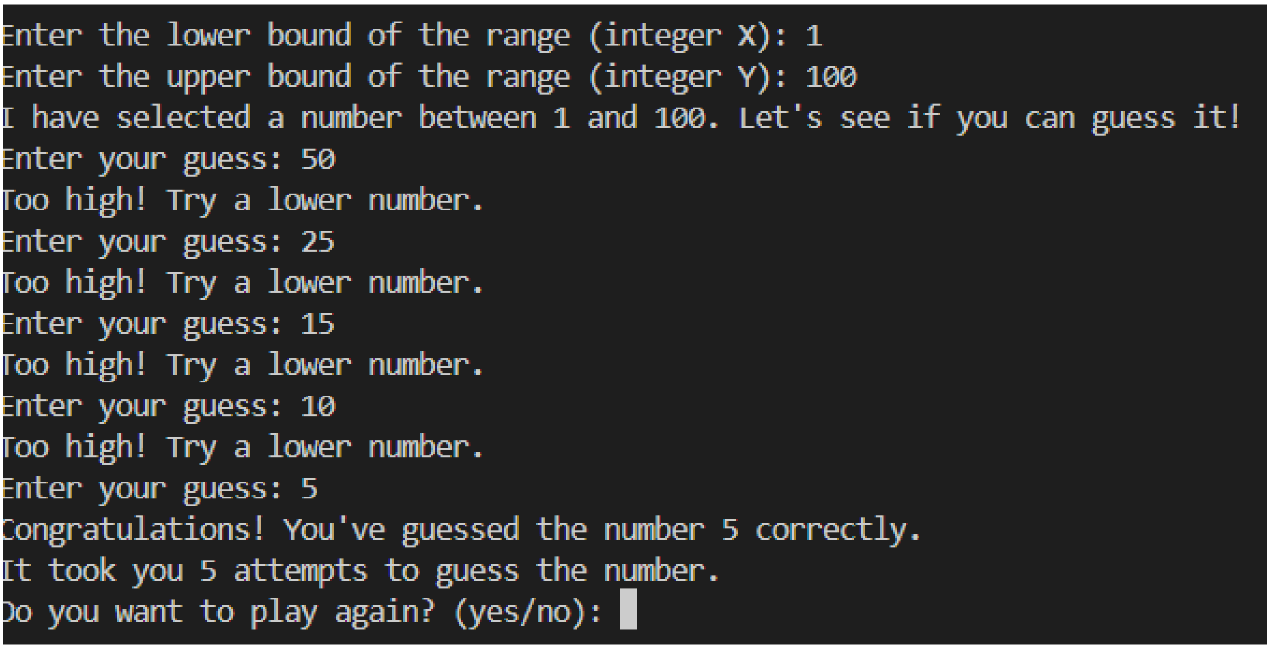
Pseudo code:

```
function number_guessing_game():
    print("Welcome to the Number Guessing Game!")
    print("===================================")

    // Input validation loop for lower and upper bounds
    while True:
        try:
            X = get_integer_input("Enter the lower bound of the range (integer X): ")
            Y = get_integer_input("Enter the upper bound of the range (integer Y): ")
            if X >= Y:
                print("Upper bound Y must be greater than lower bound X. Please try again.")
            else:
                break
        except ValueError:
            print("Invalid input. Please enter integers only.")

    // Generate a random number within the range [X, Y]
    secret_number = random.randint(X, Y)
    print("I have selected a number between", X, "and", Y, ". Let's see if you can guess it!")

    attempts = 0
    guessed = False

    // Game loop continues until the correct number is guessed
    while not guessed:
        try:
            guess = get_integer_input("Enter your guess: ")
            attempts += 1
            // Compare the guess with the secret number
            if guess < secret_number:
                print("Too low! Try a higher number.")
            elif guess > secret_number:
                print("Too high! Try a lower number.")
            else:
                guessed = True
                print("Congratulations! You've guessed the number", secret_number, "correctly.")
                print("It took you", attempts, "attempts to guess the number.")
        except ValueError:
            print("Invalid input. Please enter an integer.")

    // Ask the player if they want to play again
    play_again = input("Do you want to play again? (yes/no): ").lower()
    if play_again == "yes":
        number_guessing_game()  // Recursively start a new game
    else:
        print("Thank you for playing!")

// Helper function to get integer input from the user with error handling
function get_integer_input(prompt):
    while True:
        try:
            return int(input(prompt))
        except ValueError:
            print("Invalid input. Please enter an integer.")
```

Code for Number Guessing Game:

```python
import random
def number_guessing_game():
    print("Welcome to the Number Guessing Game!")
    print("===================================")

    while True:
        try:
            X = int(input("Enter the lower bound of the range (integer X): "))
            Y = int(input("Enter the upper bound of the range (integer Y): "))
            if X >= Y:
                print("Upper bound Y must be greater than lower bound X. Please try again.")
            else:
                break
        except ValueError:
            print("Invalid input. Please enter integers only.")

    secret_number = random.randint(X, Y)
    print(f"I have selected a number between {X} and {Y}. Let's see if you can guess it!")

    attempts = 0
    guessed = False

    # Game loop
    while not guessed:
        try:

            guess = int(input("Enter your guess: "))
            attempts += 1

            if guess < secret_number:
                print("Too low! Try a higher number.")
            elif guess > secret_number:
                print("Too high! Try a lower number.")
            else:
                guessed = True
                print(f"Congratulations! You've guessed the number {secret_number} correctly.")
                print(f"It took you {attempts} attempts to guess the number.")
        except ValueError:
            print("Invalid input. Please enter an integer.")

    play_again = input("Do you want to play again? (yes/no): ").lower()
    if play_again == "yes":
        number_guessing_game()
    else:
        print("Thank you for playing!")

number_guessing_game()
```

Output:



```
Enter the lower bound of the range (integer X): 1
Enter the upper bound of the range (integer Y): 100
I have selected a number between 1 and 100. Let's see if you can guess it!
Enter your guess: 50
Too high! Try a lower number.
Enter your guess: 25
Too high! Try a lower number.
Enter your guess: 15
Too high! Try a lower number.
Enter your guess: 10
Too high! Try a lower number.
Enter your guess: 5
Congratulations! You've guessed the number 5 correctly.
It took you 5 attempts to guess the number.
Do you want to play again? (yes/no):
```