Walkthrough: High-Low Game

Hi everyone! If you're working through this Python program and need clarity, let's walk through the code line by line so you understand exactly what's going on.

Setup: Import and Constants

```
import random
NUM_ROUNDS = 5
```

- **`import random` brings in Python's random number generator.**
- **`NUM_ROUNDS = 5` tells the game how many rounds you'll play.**

The Main Function

```
def main():
```

- **This is the main function. It's where the game begins.**

Intro Message

```
print("Welcome to the High-Low Game!")
print('--------------------------------')
```

- **Welcomes the player with a friendly message and a line to separate the header from the game.**

Keep Track of Score

```
player_score = 0
```

- Sets your score to 0 at the start. You'll earn points as you guess correctly.

The Game Loop

```
for i in range(5):
```

- The game will run 5 rounds using a `for` loop.

Inside Each Round

```
print("Round number: " + str(i+1))
```

- Shows which round you're on. `i` starts at 0, so we add 1 for human-friendly counting.

Generate Random Numbers

```
computer_num = random.randint(1,100)
player_num = random.randint(1,100)
```

- Each round, both you and the computer get a secret number between 1 and 100.

Show Player’s Number

```
print("Your number is: " + str(player_num))
```

- Your number is revealed. Now you guess if it’s higher or lower than the computer’s hidden number.

Get Your Guess

```
player_guess = input("Enter your guess: ")
player_guess = player_guess.lower()
```

- The program asks: Is your number higher or lower?
- `.lower()` lets the player type `"HIGHER" (all uppercase letters)`, `"Higher" (only one uppercase letter)`, or `"higher"(all lowercase letters)` - all are accepted.

Validate Input

```
while True:
    ...
    if player_guess == "higher" or player_guess == "lower":
        break
    else:
        print("Enter a valid input: ")
```

- This loop keeps asking you until you type either `"higher"` or `"lower"` correctly. “While True” creates an infinite loop - a loop that will run forever unless you tell it to stop.

Determine Outcome

```
if player_guess == "higher" and player_num > computer_num:
    ...
elif player_guess == "lower" and player_num < computer_num:
    ...
```

- **If your guess is right, you win the round and earn a point!**
- **Otherwise, no point for that round.**

Tell the Player

```
if player_wins_the_round:
    print("You were right! ...")
else:
    print("Your guess is incorrect! ...")
```

- **Prints a message saying if your guess was correct or not.**
- **Always shows you the computer's number.**
- **Always shows your updated score.**

After All Rounds

```
print("Thanks for playing!")
```

- Game is done! Time to wrap up and celebrate (or try again)!

Ending Messages

```
if player_score == NUM_ROUNDS:
    print("Congratulations for the perfect game! ")
```

- Check: Did the player win all rounds?
- Since `NUM_ROUNDS` is `5`, this checks if `player_score == 5`.
- If this is true, we know the player guessed correctly every single time.
- So they get a special message: “Congratulations for the perfect game!”

```
elif player_score >= NUM_ROUNDS // 2:
    print("Good job!")
```

- Check: Did the player get at least half of the answers right?
- `NUM_ROUNDS // 2` uses integer division, which ignores any decimals.
    - So `5 // 2` becomes `2` (not 2.5).
- So this condition checks if the player scored 2, 3, 4, or 5.
- If they didn’t get all 5 correct (we already checked that earlier), but got at least 2, we give them a positive message: “Good job!”

```
else:
    print("Better luck next time")
```

- If none of the above conditions are true, the score must be less than 2.
- That means the player got 0 or 1 correct guesses out of 5.
- In that case, we gently encourage them with: “Better luck next time.”

Example Outcomes

Let’s break it down further with a few examples:

`player_score`	Message Printed
`5`	Congratulations for the perfect game!
`4`	Good job!
`3`	Good job!
`2`	Good job!
`1`	Better luck next time
`0`	Better luck next time