# This **Rock, Paper, Scissors** game

This **Rock, Paper, Scissors** game is a simple Python program that allows a user to play against the computer in multiple rounds. The game tracks the score and ends when the user chooses to exit. Below is a complete breakdown of how this code works.

NAME : YOUSIF SANI

## Step-by-Step Explanation

### 1. Importing the Random Module

python Copy Edit

```python
import random
```

- The `random` module is imported to allow the computer to randomly choose between "rock", "paper", and "scissors".

### 2. Defining Choices and Initializing Scores

python Copy Edit

```python
choices = ["rock", "paper", "scissors"]

# Initialize scores
user_score = 0
computer_score = 0
```

- A list `choices` contains the three possible moves in the game.
- `user_score` and `computer_score` are initialized to `0` to keep track of wins.

### 3. Displaying a Welcome Message

```python
print("Welcome to Rock, Paper, Scissors!")
print("Type 'exit' to stop playing.")
```

- A greeting message is shown to the user.
- The game informs the user that they can exit anytime by typing `"exit"`.

### 4. Running the Game in a Loop

```python
while True:
```

- A `while` loop runs indefinitely, allowing the user to play multiple rounds.

### 5. Getting User Input

```python
user_choice = input("\nEnter rock, paper, or scissors: ").lower()
```

- The game prompts the user to enter "**rock**", "**paper**", or "**scissors**".
- `.lower()` ensures that the input is converted to lowercase, making it case-insensitive.

## 6. Exit Condition

```python
if user_choice == "exit":
    break
```

- If the user types `"exit"`, the loop breaks, ending the game.

## 8. Computer Chooses Randomly

```python
computer_choice = random.choice(choices)
print(f"Computer chose: {computer_choice}")
```

- The computer randomly selects either `"rock"`, `"paper"`, or `"scissors"`.
- The choice is displayed to the user.

## 7. Validating the User's Input

```python
if user_choice not in choices:
    print("Invalid choice! Try again.")
    continue
```

- The program checks if the user entered a valid choice.
- If not, an error message is displayed, and `continue` restarts the loop without proceeding further.

### 9. Determining the Winner

```python
if user_choice == computer_choice:
    print("It's a tie!")
```

- If both the user and the computer select the same option, it's a tie.

```python
elif (user_choice == "rock" and computer_choice == "scissors") or \
     (user_choice == "scissors" and computer_choice == "paper") or \
     (user_choice == "paper" and computer_choice == "rock"):
    print("You win this round!")
    user_score += 1
```

- The user wins if:
  - Rock beats Scissors
  - Scissors beat Paper
  - Paper beats Rock
- The user's score is increased by 1.

```python
else:
    print("Computer wins this round!")
    computer_score += 1
```

- If the conditions above are not met, the computer wins, and its score is increased.

### 10. Displaying the Current Score

```python
print(f"Score -> You: {user_score} | Computer: {computer_score}")
```

- The program prints the updated scores after each round.

### 11. Displaying Final Score Before Exiting

```python
print("\nFinal Score:")
print(f"You: {user_score} | Computer: {computer_score}")
print("Thanks for playing!")
```

- When the user exits, the final scores are displayed.
- A thank-you message is shown.