Assignment 5 Revision– Instructions

Python problems

Practical 5.1

Create a Python program that consists of two interactive games: "Guess the Number" and "Rock, Paper, Scissors."
Guess the Number: The program generates a random number between 1 and 10. The player is prompted to guess the number, which must be between 0 and 10. If the guess is too low or too high, the program provides feedback. If the guess is correct, the player wins.
Rock, Paper, Scissors: The player is prompted to choose between rock, paper, or scissors. The program randomly selects one of these choices for the computer. The choices are compared, and the winner is determined based on the classic rules of the game. If it's a tie, the game continues until there is a winner.

Create the following functions for the **Guess the Number** game:

- **generate_random_number()**: This function generates a random number between 1 and 10.
- **get_player_guess()**: This function prompts the player to enter their guess. Additionally, it continuously prompts the player to enter a guess until a valid input (an integer between 0 and 10) is provided.
- **check_guess()**: This function checks if the player's guess is too high, too low, or correct, by comparing the player's guess with the random number generated. Then it prints appropriate messages based on the comparison and returns True if the guess is correct, otherwise returns False.

Create the following function for the **Rock, Paper, Scissors** game:

- **play_rock_paper_scissors()**: This function prompts the player to input their choice, then randomly selects a choice for the computer. It then compares the choices to determine the winner and prints the outcome.

main(): This serves as the entry point of the program and manages the flow of the game:

- Main Menu: The main menu offers options to play either "Guess the Number" or "Rock, Paper, Scissors," or to exit the game. The player can select an option by entering the corresponding number. Invalid choices prompt the player to try again.
- After finishing a game (either by winning or choosing to exit), the player is returned to the main menu. They can choose to play another game or exit the program entirely.
- Implement input validation to ensure that user inputs are within the expected range or format and provide feedback for invalid inputs

Submit your Python file(*.py) here on eFundi, under this assignment, and make sure you attach and submit your file successfully. Save file as ***StudentNumber_Prac5-revision.py***

Output

```
Welcome to Guess the Number and Rock, Paper, Scissors!
Menu:
1. Guess the Number
2. Rock, Paper, Scissors
3. Exit
Enter your choice: 5
Invalid choice. Please enter 1, 2, or 3.
Menu:
1. Guess the Number
2. Rock, Paper, Scissors
3. Exit
Enter your choice: 1
Enter your guess (between 0 and 10): 100
Please enter a number between 0 and 10.
Enter your guess (between 0 and 10): 5
Too high! Try again.
Enter your guess (between 0 and 10): 3
Too low! Try again.
Enter your guess (between 0 and 10): 4
Congratulations! You've guessed the correct number.
Menu:
1. Guess the Number
2. Rock, Paper, Scissors
3. Exit
Enter your choice:
```

```
Enter your choice: 2
Enter your choice (rock, paper, scissors): rock
Computer's choice: paper
Computer wins! Better luck next time.
Menu:
1. Guess the Number
2. Rock, Paper, Scissors
3. Exit
Enter your choice: 2
Enter your choice (rock, paper, scissors): paper
Computer's choice: paper
It's a tie! Let's play again.
Enter your choice (rock, paper, scissors): rock
Computer's choice: scissors
Congratulations! You win!
Menu:
1. Guess the Number
2. Rock, Paper, Scissors
3. Exit
Enter your choice:
```

```
Menu:
1. Guess the Number
2. Rock, Paper, Scissors
3. Exit
Enter your choice: 2
Enter your choice (rock, paper, scissors): ty
Computer's choice: paper
Invalid choice. Please choose rock, paper, or scissors.
Menu:
1. Guess the Number
2. Rock, Paper, Scissors
3. Exit
Enter your choice:
```

	Rubric		
Prak 5-revision	**Functionality**		
	generate_random_number(): Generates random numbers correctly within the specified range.	**2**	
	get_player_guess(): Accepts player input and validates it correctly.	**3**	
	check_guess(): Accurately evaluates the player's guess and provides appropriate feedback.	**3**	
	play_rock_paper_scissors(): Implements the game logic correctly and determines the winner accurately.	**4**	
	User Interface		
	Displays the game outcomes and feedback in a user-friendly manner.	**1**	
	Handles invalid inputs and provides clear error messages.	**1**	
	Validates inputs to prevent out-of-range or invalid guesses.	**1**	
	Main		
	main() function effectively manages the flow of the program, allowing the player to navigate between game options.	**2**	
	Allows the player to exit the program.	**1**	
	Code Structure and Readability		
	Both programs work as intended	**1**	
	Proper use of comments for explaining the purpose of each function and major code blocks.	1	
	Total	**20**	