Logging Player Names

Create a `Player` class with the following properties:

- `String name`
- `int score`

Instructions:

1. Populate a `List<Player>` with a few players and their scores.
2. Use `forEach` with a lambda to log each player's name and score in the format: `"Player: <name>, Score: <score>"`.

Example Input:

```
List<Player> players = List.of(
    new Player("Alice", 20),
    new Player("Bob", 30),
    new Player("Charlie", 25)
);
```

Expected Output:

```
Player: Alice, Score: 20
Player: Bob, Score: 30
Player: Charlie, Score: 25
```

Removing Players with Low Scores

Using the `Player` class, write a program that removes all players with scores less than 25 using `removeIf`.

Instructions:

1. Populate a `List<Player>` with a few players and their scores.
2. Use `removeIf` with a lambda to remove players with scores below 25.

Example Input:

```
List<Player> players = new ArrayList<>(
    List.of(
        new Player("Alice", 20),
```

```
        new Player("Bob", 30),
        new Player("Charlie", 25)
    )
);
```

Expected Output:

```
[Player{name='Bob', score=30}, Player{name='Charlie', score=25}]
```

Updating Scores

Using the `Player` class, write a program that doubles the scores of all players using `replaceAll`.

Instructions:

1. Populate a `List<Player>` with a few players and their scores.
2. Use `replaceAll` with a lambda to update each player's score by doubling it.

Example Input:

```
List<Player> players = new ArrayList<>(
    List.of(
        new Player("Alice", 10),
        new Player("Bob", 15),
        new Player("Charlie", 20)
    )
);
```

Expected Output:

```
[Player{name='Alice', score=20}, Player{name='Bob', score=30},
 ↪  Player{name='Charlie', score=40}]
```

Sorting Players

Using the `Player` class, sort a list of players by their scores in descending order. If two players have the same score, sort them alphabetically by name.

Instructions:

1. Populate a `List<Player>` with a few players.
2. Use `sort` with a lambda to implement the sorting logic.

Example Input:

```
List<Player> players = new ArrayList<>(
    List.of(
        new Player("Alice", 15),
        new Player("Bob", 20),
        new Player("Charlie", 20),
        new Player("Daisy", 10)
    )
);
```

Expected Output:

```
[Player{name='Bob', score=20}, Player{name='Charlie', score=20},
↪  Player{name='Alice', score=15}, Player{name='Daisy',
↪  score=10}]
```

Adding Prefix to Player Names

Write a program that prepends "Player-" to each player's name using `forEach` and a lambda.

Instructions:

1. Populate a `List<Player>` with a few players and their scores.
2. Use `forEach` to update each player's name.

Example Input:

```
List<Player> players = new ArrayList<>(
    List.of(
        new Player("Alice", 25),
        new Player("Bob", 30),
        new Player("Charlie", 35)
    )
);
```

Expected Output:

```
[Player{name='Player-Alice', score=25},
 ↪  Player{name='Player-Bob', score=30},
 ↪  Player{name='Player-Charlie', score=35}]
```