1. **Player** class is created with Id, name, age properties (auto-implemented)
2. **ITeam** interface is created with

* **Add**
* **Remove**
* **GetPlayerById**
* **GetPlayerByName**
* **GetAllPlayers**

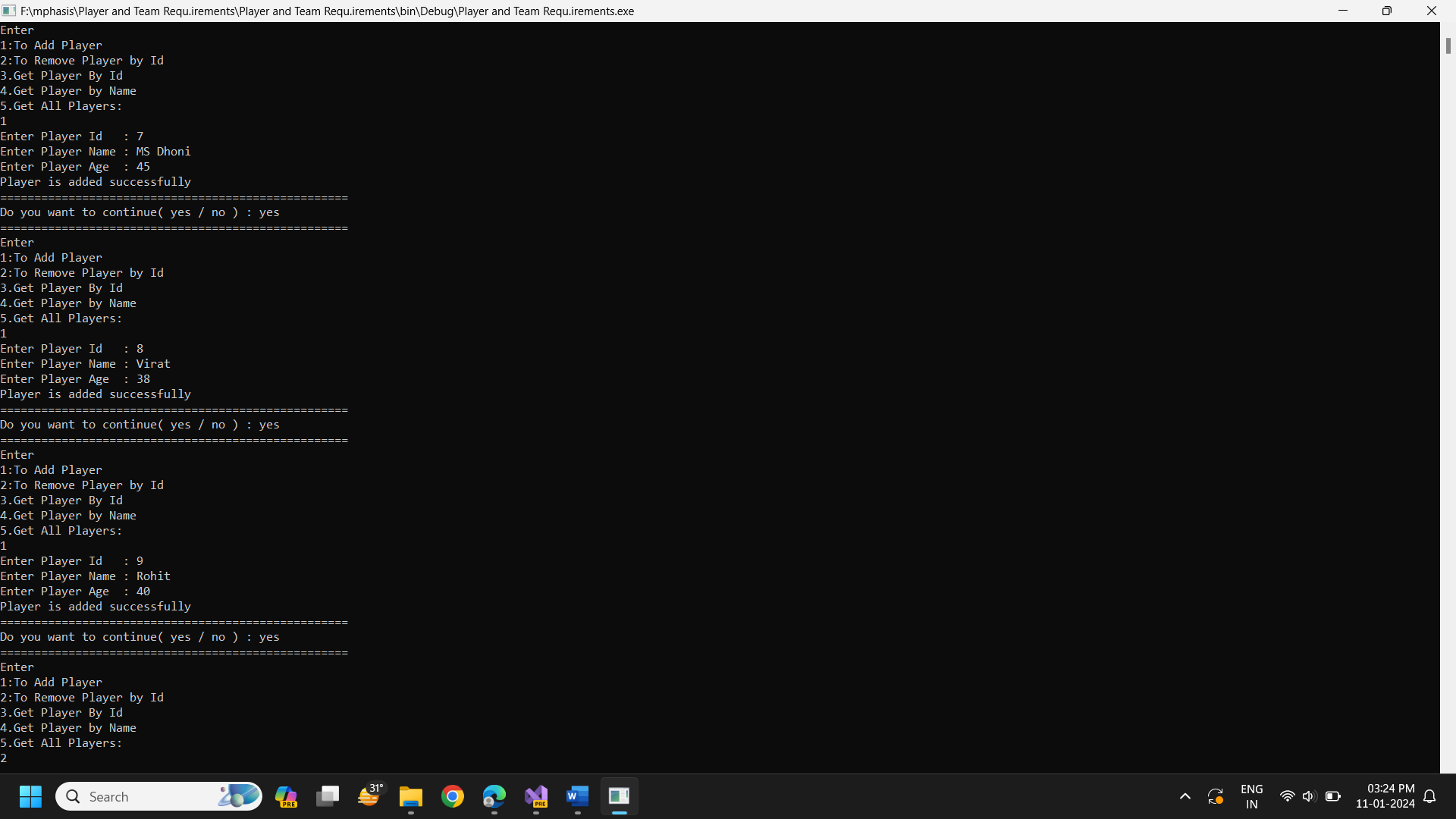
Function declarations

1. **OneDayTeam** class is created implementing **ITeam** interface. All function declarations in the interface are implemented in the class
2. A global(Static) List of Player named **oneDayTeam** is created
3. Capacity limit of the static **oneDayTeam** is set to **11**(max team members limit in cricket) and this constraint is included into the Add() function to avoid adding players beyond 11 in a team
4. In Program.cs file, menu driven approach using **switch** case for 5 operations as mentioned above in **OneDayTeam** class
5. **OneDayTeam** class in instantiated.
6. The above mentioned 5 operations are done inside switch case

* In **Add** function, **Player** parameter is sent to add it to the static list checking the capacity constraint of team players in list
* In **Remove** function, **playerId** is sent as parameter. Player Object corresponding to the Id sent is found using lambda expression and removed from the list
* In **GetPlayerById** function, **playerId** is sent as parameter. Player Object corresponding to the Id sent is found using lambda expression and details of that player object is printed
* In **GetPlayerByName** function, **playerName** is sent as parameter. Player Object corresponding to the Name sent is found using lambda expression and details of that player object is printed
* In **GetAllPlayers** function, the static List of players is returned and traversed using foreach all players details is printed

All given 5 operations are checked for its correctness from output.

* Add() and Remove() is verified for its functionality ( ***Fig 1*** )



**Figure 1**

* GetPlayerById(),GetPlayerByName(), GetAllPlayers() functionality is verified (***Fig 2***)

**Figure 2**

