Student ID - 1905188

# CE-303 – ADVANCE PROGRAMMING

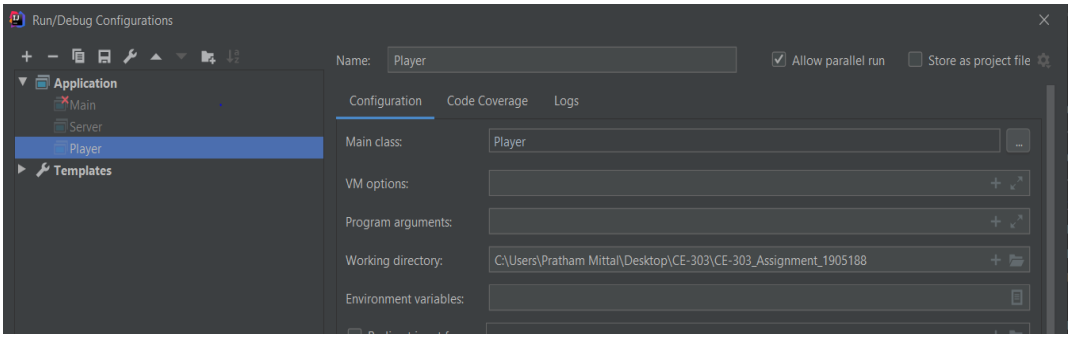
Assignment Report

# CE-303-ADVANCE PROGRAMMING

## Server & Client Assignment

## Running the Server and Client

The Java program has been developed in a way that both Server and Client can be run in one project window. To run more than one instance of client through IntelliJ the user will need to enable the “Allow parallel run” option




## Implemented Functionality

| **Function** | |
|---|---|
| Client establishes a connection with the server | Yes |
| Client is assigned a unique ID when joining the game | Yes |
| Client displays up-to-date information about the game state | Yes |
| Client allows passing the ball to another player | Yes |
| Server manages multiple client connections | Yes |
| Server accepts connections during the game | Yes |
| Server correctly handles clients leaving the game | Yes |

# Protocol

Client i.e. Player connects to socket.

Player opens Scanner and PrintWriter and the open input and output streams.

Player reads with scanner the input and imports data from Server that prints information consisting of PlayerID, Connected Players and the Player who has the ball.

Player receives data in Boolean form passed by a method that determines if the player has the ball or not.

If Player has the ball, which is determined by the information in the Server, the player program sends integer value to the server of the player ID who the ball is to be passed.

If player does not have the ball, the Boolean is false and the message is displayed.

Socket is closed upon termination.

# Client Threads

Only one thread is running in the Client side program. This is the main method thread and is responsible for displaying the game states in the client side command line and is also responsible for reading and writing data between server and PlayerHandler.

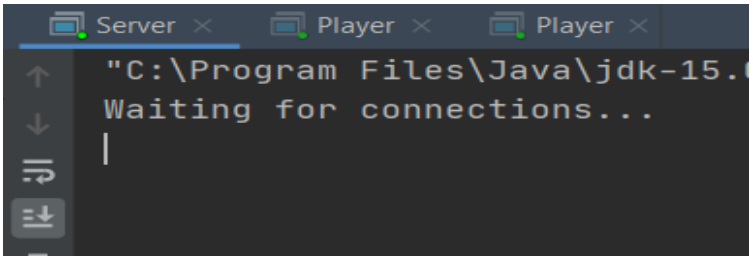
# Server Threads

Two threads are generated when running Server Class. One is generated by default by running the class and one is manually called which is a object of PlayerHandler. This helps in assigning one thread to every player in the server so that an infinite amount of players can join the game. This thread has the main task of managing all the data input and output between the player i.e. client and the server.

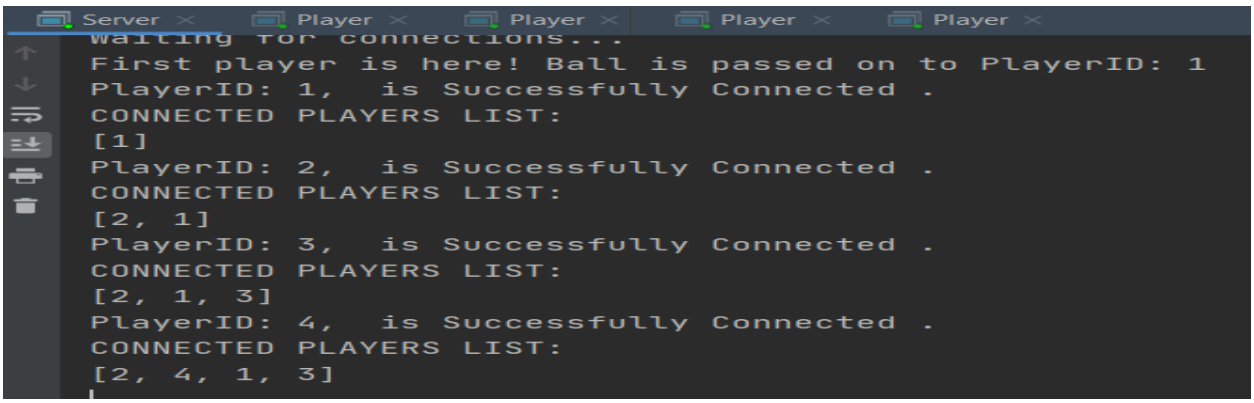
```
Thread thread = new PlayerHandler(socket, playerid);
thread.start();
```

# Command Line Output

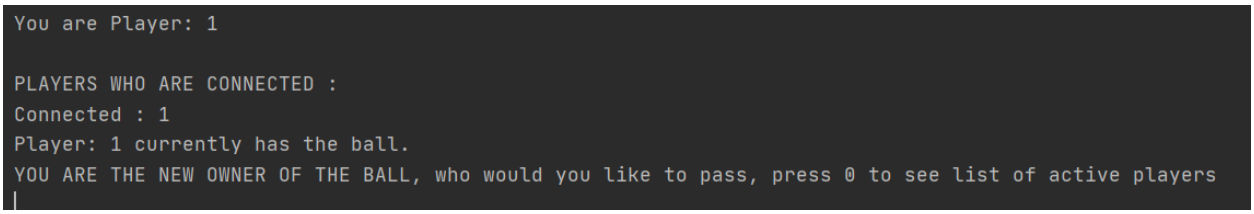
This is the output when the server is first run.



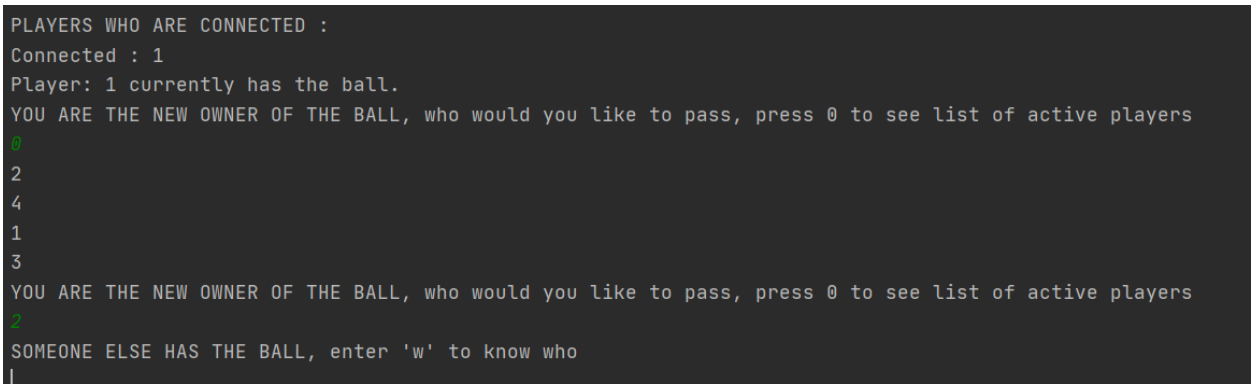
The output after connection of players look like the screen shot below, it tells if player is connected or not.



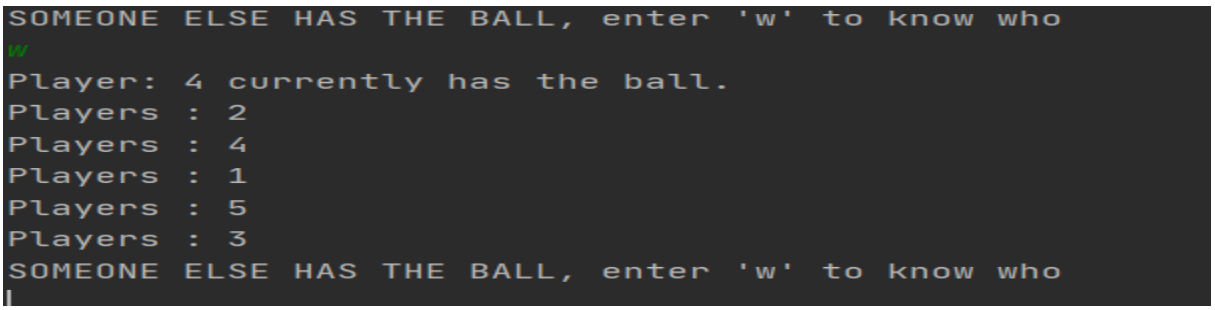

That is hat the first player sees.




Displaying the player list and passing the ball.




Other players checking the list and who has the ball, notice how the list got amended since a new player 5 joined the game.



Disconnecting player, notice how when player with ball disconnected the ball was passes to other player and the list is displayed in the server command line.

```
Player: 5,  Connection closed.
Since Player with ball disconnected, passing ball to Player: 4
2
4
1
3
```

# Project Review

I personally found this assignment very challenging at almost every stage during the development cycle. It was really interesting and new for me to program with sockets as I never thought that sockets are used and are important in program development. But now I see that how sockets are and can be used in the real world. This assignment in particular has really enhanced my programming skills and understanding about the Java language. I managed to fill every requirement in the assignment requirement table which was considerably difficult since I came across many bugs for e.g. one was that only the last player who joined can see the list when he has the ball by pressing “0” and if any other player tries to do that they got disconnected.

I am particularly proud of my ability to solve bugs after this assignment and also that I was able to fulfill each and every requirement asked by the assignment document. The program is executing everything just as I wanted the program to execute things. Also the program is handling all the errors correctly, one in major, which is to tackle error regarding wrong input. For that I am using default case in switch taught in the labs and disconnecting the player after that with a clean error message.

One thing that I could have done better with this assignment was my prerequisite knowledge about the subject and the time management. This is happened because I was more focused on my capstone project and other modules. I will definitely try to plan the development phase for a project ahead and give my self more time to gain knowledge about the topic for any other assignment that I will receive in future.