```
import java.util.Scanner; //Imports scanner for use.

/**
 * Precondition
 * The user is ready to play a mini-game.
 * User must have access to the necessary mini-games (Tic Tac Toe, Rock
Paper Scissors, Memory Game,HangMan, and Unscramble).
 * User can input responses to prompts through the console.
 * Game classes for each game are implemented with their respective
minigameplayer() methods.
 */

public class Minigame_player {
    public static void main(String[] args) {
        Scanner170 console = new Scanner170(System.in); //Initialize
Scanner170 for input (console)
        boolean keepPlaying = true; //Set's boolean variable
'keepPlaying' to true to start the game loop.

        //Loop while 'keepPlaying' variable is true.
        while (keepPlaying) { //Game selection code is shown to user in
the console, each game is assigned a number.
            System.out.println("Welcome to the mini-game Player! Choose
you're game: "); //Displays a welcome message in the console to the
user, and a list of games.
            System.out.println("1. TicTacToe");
            System.out.println("2. Rock Paper Scissors");
            System.out.println("3. Memory Game ");
            System.out.println("4. Hangman");
            System.out.println("5. Unscramble");

            System.out.print("Enter your choice: "); //Asks user to
enter their game choice (integer 1-5) that corresponds to  game.
            int choice = console.nextInt(); //Reads the user's choice.
            System.out.println("choice is " + choice); //Shows user the
choice they've made in console.

            switch (choice) {  //Based on the user's input, if the user
enters a valid integer.
                case 1: //User calls (1) TicTacToe.minigameplayer() to
play Tic Tac Toe.
                    TicTacToe.minigameplayer();
                    break;
                case 2: //User calls (2)
Rockpaperscissors_game.minigameplayer() to play Rock Paper Scissors.
                    Rockpaperscissors_game.minigameplayer();
                    break;
                case 3:  //User calls (3) Memory.minigameplayer() to
play Memory Game.
                    System.out.println("you've choosen Memory Game");
//Print out what the user has chosen to the console.
                    Memory.minigameplayer();
                    break;

                case 4:
                    HangMan.minigameplayer(); //User calls (4)
HangMan.minigameplayer() to play Hang Man.

                case 5:
                    Unscramble.minigameplayer(); //User calls (5)
Unscramble.minigameplayer() to play Unscramble.
                    break;

                default:
                    System.out.println("Invalid choice. Please select
the game you would like to play");
                    //If user gives invalid input, an invalid input
screen is shown and the user is prompted to try again.
            }

            System.out.print("Do you want to play another game?
(yes/no): "); //Ask user in console if they wish to play another game.
            String playAgain = console.nextLine(); //Read user's
response.
            keepPlaying = playAgain.equalsIgnoreCase("yes"); //If
'playAgain' = yes, set keepPlaying to true, allowing for the user to
play another round.
        } //else set 'keepPlaying' to false which will result in ending
the game loop.
        console.close(); //Closes the scanner object.
        System.out.println("Thank You For Playing! "); //Display Thank
You message to user in console.

/** Post-Conditions
 * The program is able to maintain game selection flow, handles
multiple rounds of games, and keeps track of and responds to valid user
input.
 * The user is able to choose a game from the mini-player
 * The user is able to play the game again, if they choose to through
console input.
 * If the user chooses to exit Thank You message is displayed in
console.
 * The user is explained what to input into the console to play the
game
 * The user sees a Winner/Loser or tie result for each game in the
console, or an invalid input screen.
 * The program terminates.
 */
    }
}
```