```
//imports
import java.io.FileWriter;
import java.io.IOException;
import java.util.*;

//variables to be used in different functions

public class Main {

    static void rules() {
        //SCANNER FOR USER INPUT TO EXIT RULES
        Scanner accept = new Scanner(System.in); //takes users input

        //PRINT STATEMENT FOR RULES
        System.out.println("\nTHE RULES: \n\n" +
                "1. The deck of cards contains 52 shuffled playing
cards!\n" +
                "2. When the game starts, you will be given 5 cards from
the deck, and the Computer, lets call it Connor, will take one card from
the deck.\n" +
                "3. The aim of the game is to make the ranks of the cards
add up to 11. If you do you earn a point!\n" +
                "4. The King, Queen, and Jack cards have a rank value of
10.\n" +
                "5. The Ace card has a rank value of 1.\n" +
                "6. If you are able to make 11, you can play any picture
cards in your hand to exchange them for replacement cards dealt from the
deck.\n" +
                "7. If you are unable to make 11, it's not the end of the
world! You can place a card of the same suit to continue the game.\n" +
                "8. If you are unable to make 11 or match the suits, the
game will end. :(\n" +
                "9. If you are lucky enough to use every card in the deck,
then the game will end.\n" +
                "10. The most important rule.....Enjoy the game, and thank
you for playing!\n");

        System.out.print("\n[PRESS ENTER TO CONTINUE]");
        String y;
        y = accept.nextLine();
        System.out.println("\n");
    }

    public static void main (String[] args) {
        //VARIABLES
        boolean appStart = true; //if false the game will stop running
        String selectedOption = ""; //will store the value of the selected
option
        String cU = ""; //stores current users name
        String cS = ""; //stores current score
        GameHistory gameHistory = new GameHistory(); //instance of
GameHistory to access highScores

        ArrayList <String> highScores = new ArrayList<>(); //stores all 5
high scores
        ArrayList <Integer> allHighScores = new ArrayList<>(); //ALL HIGH
SCORES WILL BE ADDED AS INTS, THEN WHEN A NEW HIGH SCORE IS SET THEN LAST
VALUE WILL BE REMOVED, THEN USER WILL BE REMOVED FROM OVERALL ARRAY, THEN
WHEN NEW USER IS ADDED IT WILL REORDER ITSELF

        //GET HIGH SCORES
```

```
        for (int i = 0; i < 5; i++){
            highScores.add(gameHistory.userScores.get(i)); //gets each user
from GameHistory and sets each value to highScores
            //System.out.println(highScores.get(i));
            String[] splitter = highScores.get(i).split(" "); //splits each
value from highScores into two different values
            //System.out.println(splitter[0] + splitter[1]);
            allHighScores.add(Integer.parseInt(splitter[1])); //adds users
score only as an int value to make it easier to compare user high scores
            //System.out.println(allHighScores.get(i));
        }


        //while loop to ensure application doesn't stop once the game has
ended
        while (appStart) { //while appStart is true the loop will continue

            /*add options (if, else if) to allow the user to select whether
            * 1. Play game (getUser() and game())
            * 2. See Rules
            * 3. See High-Scores
            * 4. Exit game and save progress (saveFile() and exit())
            * */

            Scanner option = new Scanner(System.in); //used to gather the
users input to direct them to the part they want


            System.out.println("MAKE 11\n");
            System.out.println("Hello there! Welcome to Make 11. \n" +
                    "Please select one of the following options below:");
            System.out.println("1. Play Game\n" +
                    "2. See Rules\n" +
                    "3. See High-Scores\n" +
                    "4. Save & Exit\n");
            System.out.print("\nOPTION: ");

            selectedOption = option.nextLine();

            if(selectedOption.equals("1")) { //starts game
                Scanner yN = new Scanner(System.in); //will be used to ask
user if they wish to see rules before they start.

                cU = Game.getUser(""); //runs the method getUser to get the
users name

                //WHILE LOOP TO PREVENT USER FROM ENTERING INCORRECT DATA
AND LOOSING THEIR PLACE
                Boolean uInput1 = true;
                while (uInput1) {//while user input is true/user input
invalid
                    System.out.print("\n\n Would you like to see the rules
before you play? [y/n]");
                    String input = yN.nextLine(); //gets user input

                    if((Objects.equals(input, "y")) ||
(Objects.equals(input, "Y"))){
                        cS = "0";
                        System.out.println("\nShowing Rules...");
                        rules(); //runs rules method
                        System.out.println("\nStating Game...\n");
```

```
                        cS = Game.game(""); //runs the game method
                        uInput1 = false; //sets to false to end loop
                    } else if ((Objects.equals(input, "n")) ||
(Objects.equals(input, "N"))) {cS = "0";
                        System.out.println("\nStating Game...\n");
                        cS = Game.game(""); //runs the game method
                        uInput1 = false; //sets to false to end loop
                    }else {
                        System.out.println("\nINVALID INPUT! TRY
AGAIN!\n");
                    }
                }

                for (int i = 0; i < 5; i++) {
                    int userScore = 0; //new variable for users score set
to 0
                    userScore = Integer.parseInt(cS); //converts string
value of cS(Current Score) to an int value
                    if (userScore >= allHighScores.get(i)){ //if userScore
is less than the selected index of the highest scores
                        //System.out.println("TRUE"); //test line
                        highScores.remove(4); //removes last score
                        allHighScores.remove(4); //removes last score and
name
                        highScores.add(i, cU + " " + cS); //adds new user
and score at selected index
                        allHighScores.add(i, userScore); //adds new score
at selected index
                        break; //stops for loop
                    } /*else {
                        //System.out.println("FALSE"); //test line
                    }*/
                }
                System.out.println("\n");




            } else if (selectedOption.equals("2")) { //show rules
                rules();
            }else if (selectedOption.equals("3")) { //show previous game
scores

                System.out.println("\nThe Top 5 Users Are: \n");
                for (int i = 0; i < 5; i++){
                    highScores.add(gameHistory.userScores.get(i)); //gets
each user from GameHistory and sets each value to highScores
                    System.out.println(highScores.get(i)); //prints out
user scores
                    //String[] splitter = highScores.get(i).split(" ");
                    //System.out.println(splitter[0] + splitter[1]);
                    //allHighScores.add(Integer.parseInt(splitter[1]));
                    //System.out.println(allHighScores.get(i));
                }
                System.out.print("\n");


            }else if(selectedOption.equals("4")) { //exit program

                System.out.println("\nSaving HighScores.....");
```

```
                try {
                    FileWriter writer = new
FileWriter("src/game_history.txt"); //opens file
                    for (int i = 0; i < 5; i++) {
                        writer.write(highScores.get(i) + "\n"); //writes
each user record to file
                    }
                    writer.close(); //when for loop is finished close file
                } catch (IOException e) {
                    throw new RuntimeException(e); //if try doesn't work,
it throws an error
                }


                System.out.println("\nThank you for playing!");
                appStart = false; //sets appStart to false to end
application
            } else if(selectedOption.equals("5")) { //test method for
entered name and score
                System.out.println(cU);
                System.out.println(cS);
            } else {
                System.out.println("\nINVALID INPUT! PLEASE TRY AGAIN!\n");
            }
        }
    }
}

/*NOTES

* */
```