

# ICS4U Poker Solitaire: Top Player Tips

- 1) Create a `Player` class to keep track of each Player's information (name and score).
  - a. The `Player` class should include a constructor, `compareTo` method (so that we can sort the Players by score) and a `toString` method (used to display Player info).
  - b. You will also need to make the `Player` class `Comparable` (for sorting) and `Serializable` (so that you can write the `Player` info to a file).
- 2) In the main `PokerSolitaire` program you will need to add code to keep track of a List of the top Players.

- a. Create a variable for the List at the top of the program with the other instance variables.

```
private ArrayList<Player> topPlayers;
```

- b. In the `PokerSolitaire` constructor, you will read in the file of top Players. If no file is available, we will create a new empty `ArrayList` for the `topPlayers`. We can use a `try... catch` block to check if there is a file. Here is some sample code.

```
try
{
    // Try to open the file and read in the top player information
    // Read the entire ArrayList from a file
    ObjectInputStream fileIn =
        new ObjectInputStream(new FileInputStream("topPlayers.dat"));
    topPlayers = (ArrayList<Player>)fileIn.readObject();
    fileIn.close();
}
catch (Exception) // This could include different types of Exceptions
{
    // If we had trouble reading the file (e.g. it doesn't exist) or
    // if our file has errors an Exception will be thrown and we can
    // create a new empty list
    topPlayers = new ArrayList<Player>();
}
```

- c. If the `topPlayers` List changes, we will need to write the List to a File. This code will also throw Exceptions so you may want to include a `try catch` block. When you get an error, use the hint to help deal with this error.

```
// Write the entire ArrayList to a file
ObjectOutputStream fileOut =
    new ObjectOutputStream(new FileOutputStream("topPlayers.dat"));
fileOut.writeObject(topPlayers);
fileOut.close();
```

- d. At the end of each game, if the score is high enough to make the `topPlayers` List, you will need to create a new `Player` and add them to the `topPlayers` List. You will also want to sort the `topPlayers` List to keep it in order (highest score first).
  - e. To display the List you will need to create a new dialog box. You can create a new `String` (use a `StringBuilder` to build it) with all of the data for all of the Players in the `topPlayers` list (use a `for` each loop to go through the List and `"\n"` to get new lines). Here is the code to create the dialog box. It assumes `playerData` is a `String` variable that holds all of the `Player` data.

```
JOptionPane.showMessageDialog (cardArea, playerData,
                                "Top Scores", JOptionPane.INFORMATION_MESSAGE);
```