FAULT DETECTION AND CORRECTION

In the given project there are a lot of seeded faults and we have corrected them one by one which makes the code executable. There is atleast a single fault in every testable method and the corrections are shown as below:

1. First fault is in the method AddGame(String name, int maxPlayers)
   Here the first fault is in the constructor of Game.

   This is Line number 39.

   Here Game is called as Game game = new Game(name,maxPlayers + 1)
   But the user input is only maxPlayers.

   This is because maxPlayers + 1 will make the game associated with an extra player in the constructor while in the method the game is associated with a less player. So, this is a fault and must be made to this.

   Fault: Game game = new Game(name,maxPlayers + 1)
   Correction: Game game = new Game(name,maxPlayers)

2. Second fault is in line number 45.
   Here we do gameAssociationList[gameCounter++] = association;

   The fault is that when we do gameCounter++ here and again after that in the next line we do gameCounter++ so this updates gameCounter twice hence gameAssociationList does not store the correct list of games in the correct order that is goes from 1 to 3 to 5 and so on. So, when we use this method anywhere else it gameAssociationList starts from 0 to 1,2,3 and so on. So, the games are not stored in the correct position, hence will lead to an error and failure. Hence, we need to change this.

   Fault: gameAssociationList[gameCounter++] = association. //line no 45
   Correction: gameAssociationList[gameCounter] = association. // line no 45

3. The third fault is in the next method searchGame(String name)
   The fault is in the line number 58.
   Here in the method which searches for the respective game by getting a random user input which is also a game. So here in this method in line no 58 we do this:

   Game storedGame = gameList[i+1];

   Here the problem is we start searching from position 1 when we must start searching from position 0. This is because we are storing the first game in position 0 but searching from position 1. So, say Cricket is stored in position 0 but we are searching from position 1 and position 1 could be another sport. Hence this will lead to failure hence this is a fault and must be changed. Basically, it will skip the first position of the array index and start searching. So, we can correct this by

Fault: Game storedGame = gameList[i+1]; // line no 58
Correction: Game storedGame = gameList[I];// line no 58

4. The next fault is in the method addPlayer(String name,String[] gameNames)

Here in line number 79: for(i=1;i<gameNames.length;i++)

It starts searching from the second array index instead of the first one as i=1.So when we send gameNames this won't check for gameNames[0] as in the next line inside the for loop it is

String gameName = gameNames[i]. So, it won't check for i=0 which is a fault. Hence, we must correct this.

Fault: for(i=1;i<gameNames.length;i++) // line no 79
Correction: for(i=0;i<gameNames.length;i++) // line no 79

5. Here the next fault in the next method
String addSchedule(String dayName, String[] gameNames)

This is in line number 108 which is for(i=0;i<gameNames.length-1;i++)
Here the fault is that if gamesNames of length 1 then the for would not get executed hence gamesPlayed would not be assigned and will to lead to an error and failure. Hence, we must correct this. 2 methods are there

Method 1:

Error: for(i=0;i<gameNames.length-1;i++) //line no 108
Correction: for(i=0;i<gameNames.length;i++) //line no 108

Method 2:

Error: for(i=0;i<gameNames.length-1;i++) //line no 108
Correction: for(i=0; i<=gameNames.length-1;i++) //line no 108

6. Here the next fault in the method
public Player searchPlayer(String name)

This is in line number line 143: for (int i=0; i < playerCounter-1; i++)

Here again as in the previous fault again when the playerCounter is updated only once when addPlayer is called so PlayerCounter is 1. So, in searchPlayer when playerCounter=1 and when it goes into the for loop the playerCounter-1 will be 0 but we are doing i=0;i<0.So this is a fault and the execution will come out of the for loop and will not go into the for loop. Hence here is where we must do the correction here.

Fault: for(int i=0; i < playerCounter-1; i++) // line no 143
Correction: for(int i=0; i < playerCounter; i++) // line no 143

Or

Fault: for(int i=0; i < playerCounter-1; i++) // line no 143
Correction: for(int i=0; i < =playerCounter-1; i++) // line no 143

7. The next fault is again in the method
public Player searchPlayer(String name)

Here the fault is in line number 145.Here the the fault is in the if condition
if(storedPlayer.name.equals(storedPlayer.name))

Here the problem Is it the values compared are both the same and it compares with itself rather than the user input. So, this will always be same. So, it's not a proper if condition.
So, we must correct this:

Fault: if(storedPlayer.name.equals(storedPlayer.name)) // line no 145
Correction: if(storedPlayer.name.equals(name)) // line no 145

8. The next fault is again in the method
public DaySchedule searchDay(String name)

Here the fault is in line number 158. Here the problem is in the for loop. Here again as in many other previous faults the problem is in the for loop.

for(int i=1; i < scheduleCounter; i++)

Here the problem is the loop is initialised in the wrong way that is I is starting from 1 rather than 0 because when addDaySchedule gets called scheduleCounter becomes 1 and in searchDay() the for loop does not function work properly as i=1; i<scheduleCounter after this condition the control does not go into the for loop and will come out of the loop. Hence this is the problem and we must correct it.

Fault: for(int i=1; i < scheduleCounter; i++) // line no 158
Correction: for(int i=0; i < scheduleCounter; i++) // line no 158

9. The fault is again in the same method as above.
DaySchedule storedDay = scheduleList[i-1]; // line no 159

The problem here is again the array initialisation. Here the problem is once for starts from i=0, and when it goes into the loop scheduleList[i-1] will become scheduleList[-1] which

is wrong as the basic concept of array tells us that there is no -1 postion. Hence this is the fault, and this must be corrected.

Fault: DaySchedule storedDay = scheduleList[i-1]; // line no 159
Correction: DaySchedule storedDay = scheduleList[i]; // line no 159

10. The fault is in the method

public String displayDayWiseSchedule(String dayName)

The problem here is after for each loop. This is at line no 210.That is that for each loop is for the iterates the no of games played. Here after the for each loop it directly appends but the issue is the gamesPlayed can be null so here we append even if it is null which is wrong. Then the object is null and there is no condition which checks it. There must be a condition which tests this which is

Fault: (Game g: gamesPlayed) { // line no 210
sb.append("Game = "+g.name);

Correction: (Game g: gamesPlayed) { // line no 210
If (g == null)
break;

11. The fault is in the method

public String displayPlayerWiseSchedule(String playerName)

The problem here is after for each loop. This is at line no 256.That is that for each loop is for the iterates the no of games played. Here after the for each loop it directly appends but the issue is the gamesPlayed can be null so here we append even if it is null which is wrong. Then the object is null and there is no condition which checks it. There must be a condition which tests this which is

Fault: (Game g: gamesPlayed) { // line no 256
sb.append("Game = "+g.name);

Correction: (Game g: gamesPlayed) { // line no 256
If (g == null)
break;

These are the faults and their respective corrections which made the code run successfully and made us achieve 97.9 code coverage.