QLC-2.4) Stubs to Mocks

```
   @Test
   public  void
find_heighest_score_with_array_of_many_return_array_of_many_using_mocks(
)
   {
      //[{“Physics”, { 56, 67, 45, 89} }, {“Art”, { 87, 66, 78} },
{“Comp Sci”, { 45, 88, 97, 56} }]
      // Arrange
      int[] physics_scores = { 56, 67, 45, 89 };
      String physics = "Physics";
      int[] art_scores = { 87, 66, 78 };
      String art = "Art";
      int[] compSci_scores = { 45, 88, 97, 56 };
      String compSci = "Comp Sci";
      ArrayList<TopicScores> topicScores = new ArrayList<>();
      topicScores.add(new TopicScores(physics, physics_scores));
      topicScores.add(new TopicScores(art, art_scores));
      topicScores.add(new TopicScores(compSci, compSci_scores));

      // Use a mock version of HighestNumberFinder
      IHighestNumberFinder hnf = mock( com.s2s.demos.findhighestnumber.
fin.HighestNumberFinder.class );
      // Setup the expectations
      when(hnf.findHighestNumber(physics_scores)).thenReturn(89);
      when(hnf.findHighestNumber(art_scores)).thenReturn(87);
      when(hnf.findHighestNumber(compSci_scores)).thenReturn(97);

      TopicManager cut = new TopicManager(hnf);
      ArrayList<TopicTopScore> expectedResult = new ArrayList<>();
      expectedResult.add( new TopicTopScore(physics, 89));
      expectedResult.add( new TopicTopScore(art, 87));
      expectedResult.add( new TopicTopScore(compSci, 97));

      // Act
      ArrayList<TopicTopScore> result = cut.findTopicHighScores
(topicScores);

      // Assert
      assertThat(expectedResult, is(result));
   }
```

For line 34 to work, we have added the equals operator to `TopicTopScore`

```
package com.s2s.demos.topicmanager;

public class TopicTopScore
{
   private  String topicName;
   private  int topScore;

   public TopicTopScore(String topicName, int score)
   {
      this.topScore = score;
      this.topicName = topicName;
   }

   public String getTopicName()
   {
      return topicName;
   }

   public int getTopScore()
   {
      return topScore;
   }

   @Override
   public boolean equals(Object anObject)
   {
      TopicTopScore rh = (TopicTopScore)anObject;

      return (topicName.equalsIgnoreCase(rh.topicName) && (topScore==rh.
topScore));
   }
}
```