Question 1 (9 marks):

A generic `Event` class is defined as follows: **class** Event { **private** String eventID; **private int** numberOfRaces; **private** Race[] races; **private** Race finals; **public** Event(String ID, int numberOfRaces) { eventID = ID; races = **new** Race[numberOfRaces]; **for** (**int** i = 0; i < numberOfRaces; i++) { races[i] = **new** Race(); } finals = **new** Race(); } **public void** addSwimmers() { // fills the qualifying heats with swimmers } **public void** fillFinals() { // fills the finals race with the best 8 from the qualifying heats } // more methods() } The EventEvent class above assumes that the event has more than 8 swimmers and requires qualifying heats. However, an event with less than 9 swimmers has no qualifying heats, so the original EventEvent class was inherited by a new class FinalsOnlyEventFinalsOnlyEvent. The same method identifier `addSwimmers` is used in both classes `Race` and `Event`. Explain why this does not cause a conflict. [3] a. Outline **two** advantages of the OOP feature "inheritance". [4] b. Outline how method overriding can help to create the new class `FinalsOnlyEvent`. [2] c.