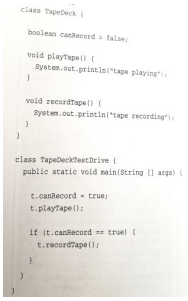
# Set 04

1 The code on the right ...



```
class TapeDeck {

  boolean canRecord = false;

  void playTape() {
    System.out.println("tape playing");
  }

  void recordTape() {
    System.out.println("tape recording");
  }
}

class TapeDeckTestDrive {
  public static void main(String [] args) {

    t.canRecord = true;
    t.playTape();

    if (t.canRecord == true) {
      t.recordTape();
    }
  }
}
```

- A is just fine!
- B does not use good practices of naming conventions
- ☑ has a problem: reference variable t was not instantiated
- D has a problem: TapeDeckTestDrive cannot access TapeDeck's variables

2 class Movie { String title, genre; int rating; } Is it possible to do: Movie m = new Movie();

- ☑ Yes, why not?
- B No, the class Movie does not have a constructor
- C No, the class Movie does not have any methods
- D No, the class Movie does not inherit from Object

3 TDD (Test-Driven Development) advocates that ...

- ☑ Tests should be written before you actually write the solution
- B Developers should focus more on getting the code right without testing
- C Tests should be postponed when the final version of the software is handled to the users
- D Test is an optional part of the development process

4 Once a class pass all of their unit tests it is correct to say that ...

- A the class is free of errors
- B the class is free of errors as long as the tests are thorough
- C the class is free of errors as long as the tests are thorough AND there aren't any unforeseen logic errors
- ☑ the class is free of errors as long as the tests are thorough AND there aren't any unforeseen logic errors AND any other problems related to how the class behaves with other parts of the code