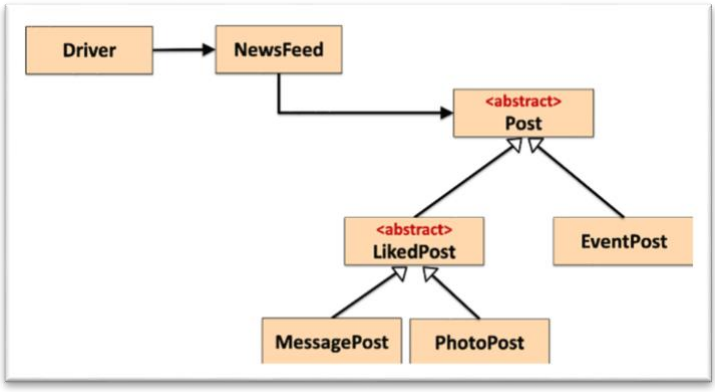
Class Diagram

All questions in this exam are based on one app.



Post
- Post(String)
- displayCondensed(): String
- getAuthor(): String
- setAuthor(String): void
- display(): String
- author: String = ""

LikedPost
- LikedPost(String)
- displayCondensed(): String ↑Post
- getLikes(): int
- setLikes(int): void
- likeAPost(): void
- unlikeAPost(): void
- display(): String ↑Post
- likes: int = 0

MessagePost
- MessagePost(String, String)
- displayCondensed(): String ↑LikedPost
- getMessage(): String
- setMessage(String): void
- display(): String ↑LikedPost
- message: String = ""

PhotoPost
- PhotoPost(String, String, String)
- displayCondensed(): String ↑LikedPost
- getCaption(): String
- setCaption(String): void
- setFilename(String): void
- getFilename(): String
- display(): String ↑LikedPost
- caption: String = ""
- filename: String = ""

EventPost
- EventPost(String, String, double)
- displayCondensed(): String ↑Post
- getEventName(): String
- setEventName(String): void
- getEventCost(): double
- setEventCost(double): void
- display(): String ↑Post
- eventName: String = ""
- eventCost: double = 0

You may detach this page from the booklet.

Name	Student Number	Course

Instructions:

- **2 hour exam.**
- **4 Questions, answer all.**
- **Fill in your name, student number and course above.**
- **Complete the areas in this booklet and give it to your invigilator before leaving the room.**
- **Except for the front page, don't tear pages from this booklet; submit as a complete booklet.**
- **You can write anywhere in this booklet and use the back of pages for additional code, rough work, etc.**

Question 1 - Models:

With this exam, you are given an inheritance hierarchy and the structure (UML) for the classes in the hierarchy. For the Post, LikedPost and MessagePost classes, code the following:

- The class definition
- The constructor
- The toString() method

Note: you do not have to define the fields or write the getters/setters.

Post.java

LikedPost.java

MessagePost.java

Question 2 - Controllers:

The class diagram has a NewsFeed class. In this class, there is an ArrayList<Post> defined:

```
public class NewsFeed {

    private ArrayList<Post> posts;

    public NewsFeed() {
        posts = new ArrayList<Post>();
    }
}
```

Each box below has a comment in it. Code each method as defined in the comment.

NewsFeed.java

```
//  Write a public method, listAllPosts(). The return type is String.
//  This method returns a list of the posts stored in the array list.
//  Each post should be on a new line and should be preceded by the
//  index number e.g.
//        0: Post 1 Details
//        1: Post 2 Details
//  If there are no posts stored in the array list, return a string
//  that contains "No Posts".
```

NewsFeed.java

```
// Write a public method, numberOfMessagePosts().  The return type
// is int.  This method returns the number of MessagePost objects
// stored in the array list.
// If there are none stored, zero is returned.
```

NewsFeed.java

```
// Write a public method, listAllPostsByAuthor(String author).
// The return type is String.
//
// This method returns a list of the posts written by an author that
// are stored in the array list (i.e. that match the parameter value).
//
// Each matching post should be on a new line and should be preceded
// by the index number e.g.
//        1: Post 2 Details
//        4: Post 5 Details
//
// If there are no posts stored in the array list, return a string
// that contains "No Posts".
//
// If there are no posts matching the author, the return string should
// have "No posts for that author".
```

NewsFeed.java

```
// If you included the sort functionality in your submitted code
// base, please write the code to sort the array list by any criteria
// you wish e.g. author name ascending.
```

Question 3 – Junit Tests:

This JUnit test class for NewsFeed contains test fixture data:

```
public class NewsFeedTest {

    private MessagePost janeMessage1, joeMessage1;

    private PhotoPost janePhoto1, joePhoto1;

    private NewsFeed newsFeedPopulated = new NewsFeed();
    private NewsFeed newsFeedEmpty = new NewsFeed();

    @BeforeEach
    void setUp() {

        //Creating four post objects, 2 MessagePost and 2 PhotoPost.
        janeMessage1 = new MessagePost("Jane Doe", "Jane's first post");
        joeMessage1 = new MessagePost("Joe Soap", "Joe's first post");
        janePhoto1 = new PhotoPost("Jane Doe", "Jane's Profile", "jane.jpg");
        joePhoto1 = new PhotoPost("Joe Soap", "Joe's Profile", "joe.jpg");

        //adding the four objects to the newsFeedPopulated object
        newsFeedPopulated.addPost(janeMessage1);
        newsFeedPopulated.addPost(joeMessage1);
        newsFeedPopulated.addPost(janePhoto1);
        newsFeedPopulated.addPost(joePhoto1);

        // Note that no posts are added to the newsFeedEmpty object
    }
}
```

Write the two test methods that are specified in the box below.

Note:

- You can assume that you don't have to write import statements.
- You can name the test methods anything you wish.

NewsFeedTest.java

In a previous question, you were asked to write a method called numberOfMessagePosts().

- This method returned the number of MessagePost objects stored in the array list.
- If there were none stored, zero is returned.

Using the test fixture data given above, write the two test methods, detailed below.

Test Method 1: This test method should verify that zero is returned by numberOfMessagePosts() when no message posts are stored in the NewsFeed object.

Test Method 2: This test method should verify that the correct number of message posts is returned by numberOfMessagePosts() when message posts are stored in the NewsFeed object.

Question 4: Driver:

The class diagram has a Driver class. In this class, there is an object of the NewsFeed class.

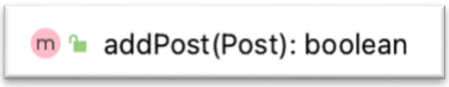
```
public class Driver {

    private NewsFeed newsFeed = new NewsFeed();

    public static void main(String[] args) {
        new Driver();
    }
}
```

This class displays the menu of options that the user can choose from:

In the box below, complete the code for the "Add a Post" menu option.

Use the ScannerInput class for reading from the console:

The add method that you will be calling in NewsFeed is this format:



Driver.java

```
private void addPost(){
    int option = ScannerInput.readNextInt("""
                 --------------------------
                 |   1) Add a Message Post |
                 |   2) Add a Photo Post   |
                 |   3) Add an Event Post  |
                 --------------------------
                 ==>> """);

    switch (option) {
        case 1 -> {
            // TODO Write the code to read in the data for a
            //      Message Post and add it to the NewsFeed.









        case 2 -> {
            // TODO Write the code to read in the data for a
            //      Photo Post and add it to the NewsFeed.









        }
        case 3 -> {
            // TODO Write the code to read in the data for an
            //      Event Post and add it to the NewsFeed.









        }
        default -> System.out.println("Invalid option: " + option);
    }
}
```