1. Read the following description of SportsMania and answer all the questions.

"SportsMania is a fantastic sports site for sport lovers. They cover almost all kinds of sports and provide the latest news, information, matches scheduled dates, information about a particular player or a team. Now, they are planning to provide live commentary or scores of matches as an SMS service, but only for their premium users. Their aim is to SMS the live score, match situation, and important events after short intervals. As a user, you need to subscribe to the package and when there is a live match you will get an SMS to the live commentary. The site also provides an option to unsubscribe from the package whenever you want to.

As a developer, the SportsMania asked you to provide this new feature for them. The reporters of the SportsMania will sit in the commentary box in the match, and they will update live commentary to a commentary object. As a developer your job is to provide the commentary to the registered users by fetching it from the commentary object when it's available. When there is an update, the system should update the subscribed users by sending them the SMS."

a. Identify one design pattern that you can use to provide the new feature required by SportsMania.

Observer Design Pattern

(1 mark)

b. With reference to GoF's categorization of design patterns, the design pattern you stated in Question 1.a. falls into which category of pattern? State also whether the category is object or class scope.

(2 marks)

The category is of object scope.

c. Justify why you choose the design pattern in Question 1.a.

(1 mark)

The Observer design pattern is the perfect ~~match~~ design for this type of system because the system provides ~~latest~~ contents that are constantly updated.
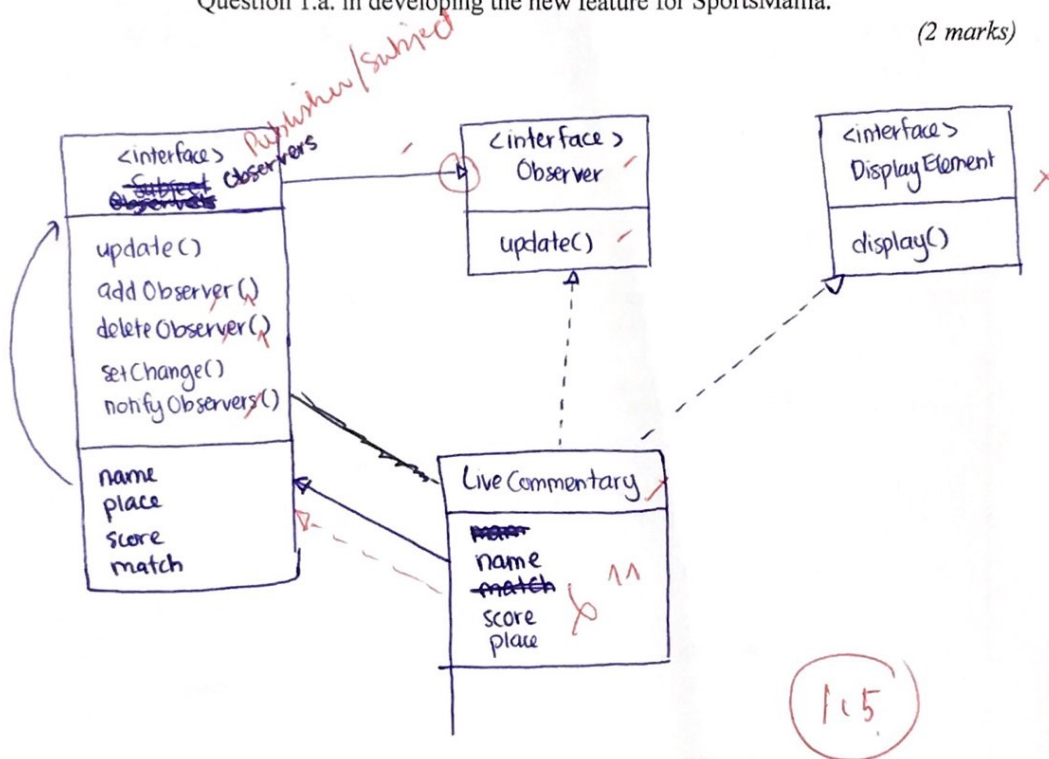
0.5

~~Latest News~~

d. Draw a UML class diagram to show how you apply the design pattern in Question 1.a. in developing the new feature for SportsMania.

*(2 marks)*

e. Create a Netbeans project named "yourMatricNumber-Test-Q1e" in the local drive and save all your files in this project folder. Type your name and matrix number on each Java file you are going to submit.

Implement the new feature for SportsMania based on your design in Question 1.d. The output should be as shown below.

Sample output:

```
Subscribing Adam Warner [New York] to Soccer Match [30th Oct 2018] ...
Subscribed successfully.

Subscribing Tim Ronney [London] to Soccer Match [30th Oct 2018] ...
Subscribed successfully.


[Adam Warner [New York]]: Welcome to live Soccer match
[Tim Ronney [London]]: Welcome to live Soccer match


[Adam Warner [New York]]: Current score 0-0
[Tim Ronney [London]]: Current score 0-0


Unsubscribing Tim Ronney [London] to Soccer Match [30th Oct 2018] ...
Unsubscribed successfully.


[Adam Warner [New York]]: It's a goal!!


[Adam Warner [New York]]: Current score 1-0


Subscribing Marrie [Paris] to Soccer Match [30th Oct 2018] ...
Subscribed successfully.


[Adam Warner [New York]]: It's another goal!!
[Marrie [Paris]]: It's another goal!!


[Adam Warner [New York]]: Half-time score 2-0
[Marrie [Paris]]: Half-time score 2-0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Assume that these represent the two subscribers receiving the commentary via SMS when the commentary object was updated with the commentary "Welcome to live Soccer match".

Assume that these represent the two subscribers receiving the commentary via SMS when the commentary object was updated with the commentary "Current score 0-0".
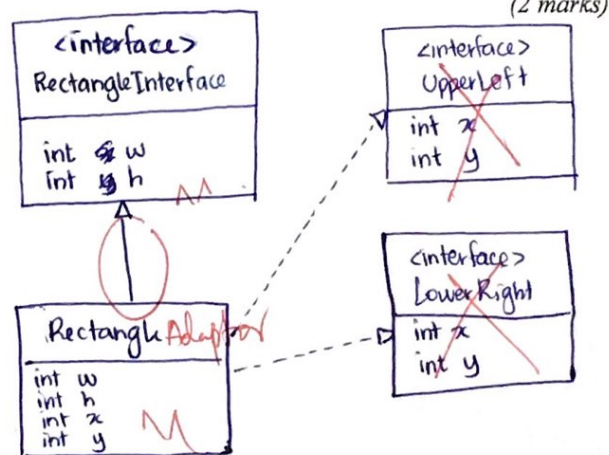
*(8 marks)*

2. A legacy *Rectangle* has a *display* method that expects to receive "x, y, w, h" parameters to draw a rectangle, where x and y represent the upper left x and y coordinates of the rectangle, w represents the width of the rectangle and h represents the height of the rectangle. However, the client wants to pass "upper left x and y coordinates" and "lower right x and y coordinates" of a rectangle to the *display* method to draw the rectangle.

a. Identify one design pattern that you can use to solve the incongruity above.

(1 mark)

Strategy Design Pattern

b. Draw a UML class diagram to show how you apply the design pattern in Question 2.a. to solve the problem.

(2 marks)



```
<interface>
RectangleInterface

int w
int h
```

```
<interface>
UpperLeft

int x
int y
```

```
<interface>
LowerRight

int x
int y
```

```
Rectangle Adaptor

int w
int h
int x
int y
```

0.5

c. With reference to GoF's categorization of design patterns, the design pattern you applied in Question 2.b. falls into which category of pattern? State also whether the category is object or class scope.

(3 marks)

-----The End-----

0.5