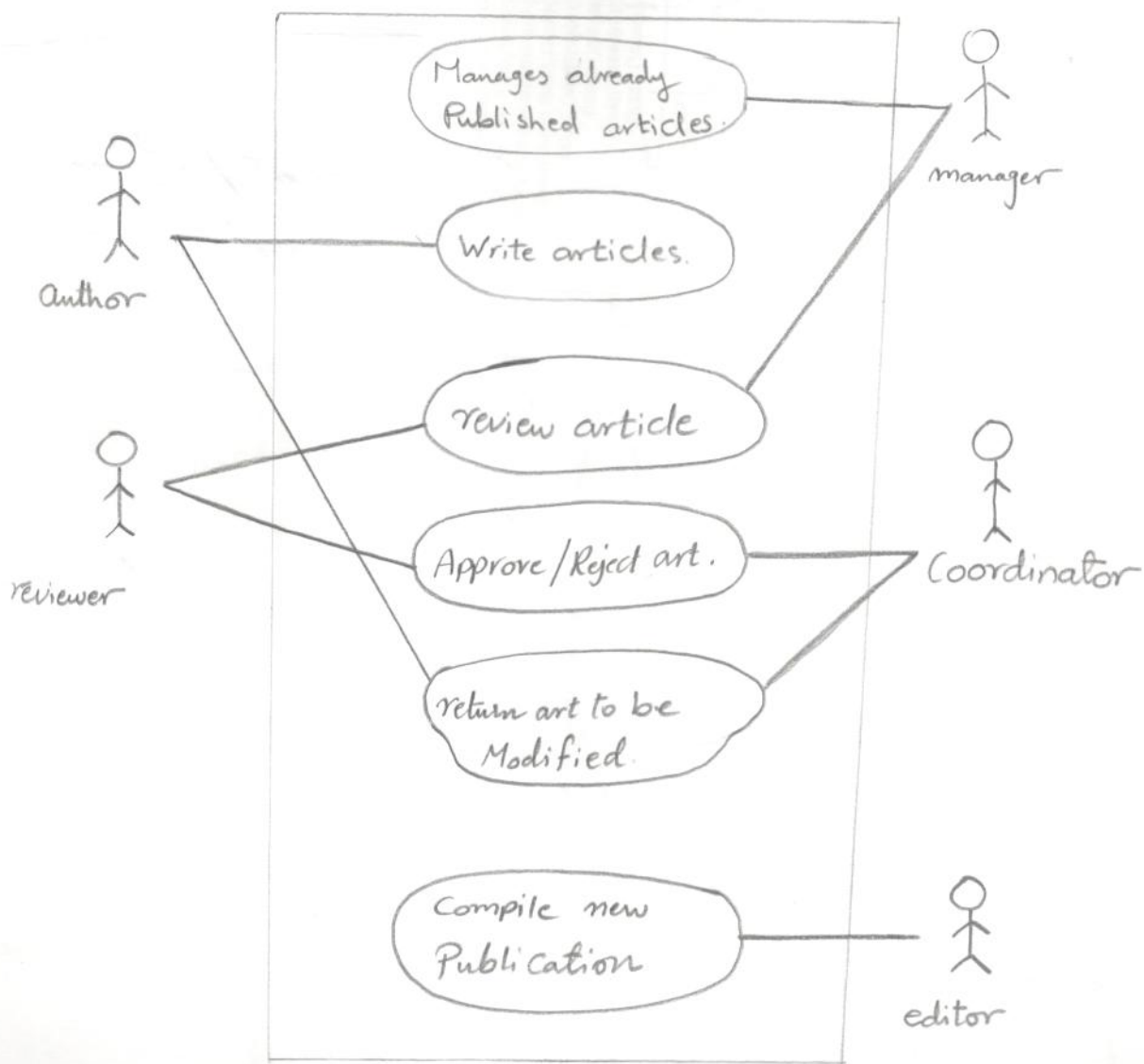Names :- • Omar Mohammed Abdebasset. (42)
• Sherif Hassan Waly (30)
• Mohammed Fathy Abdelfattah (60)
• Marwan Alaa Mahmoud (65)

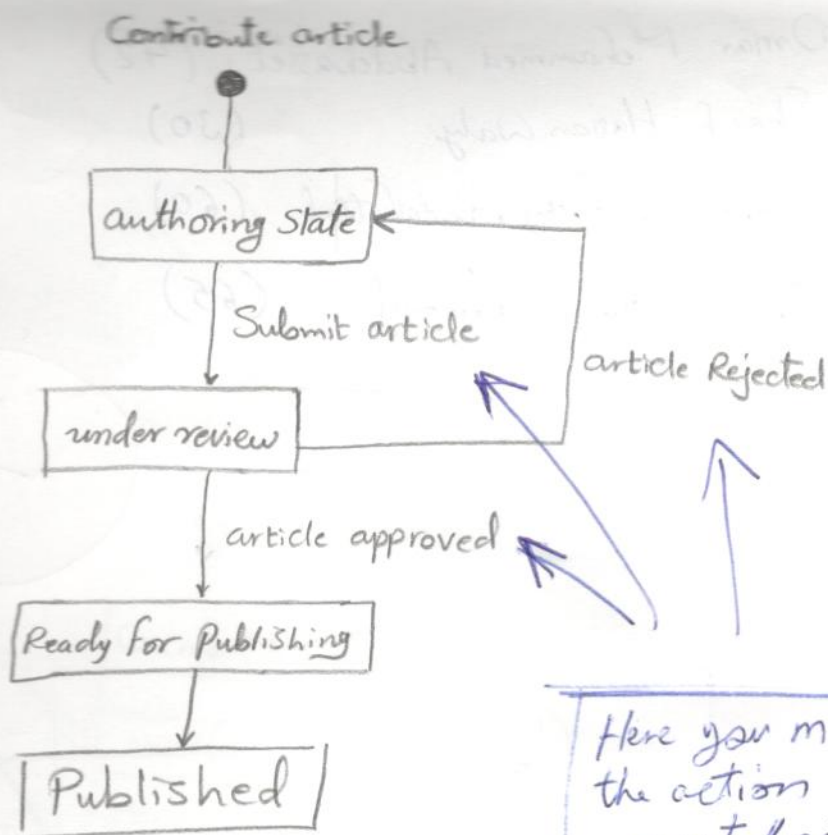• Final (2007)

Ⓐ

* Question 1 :-
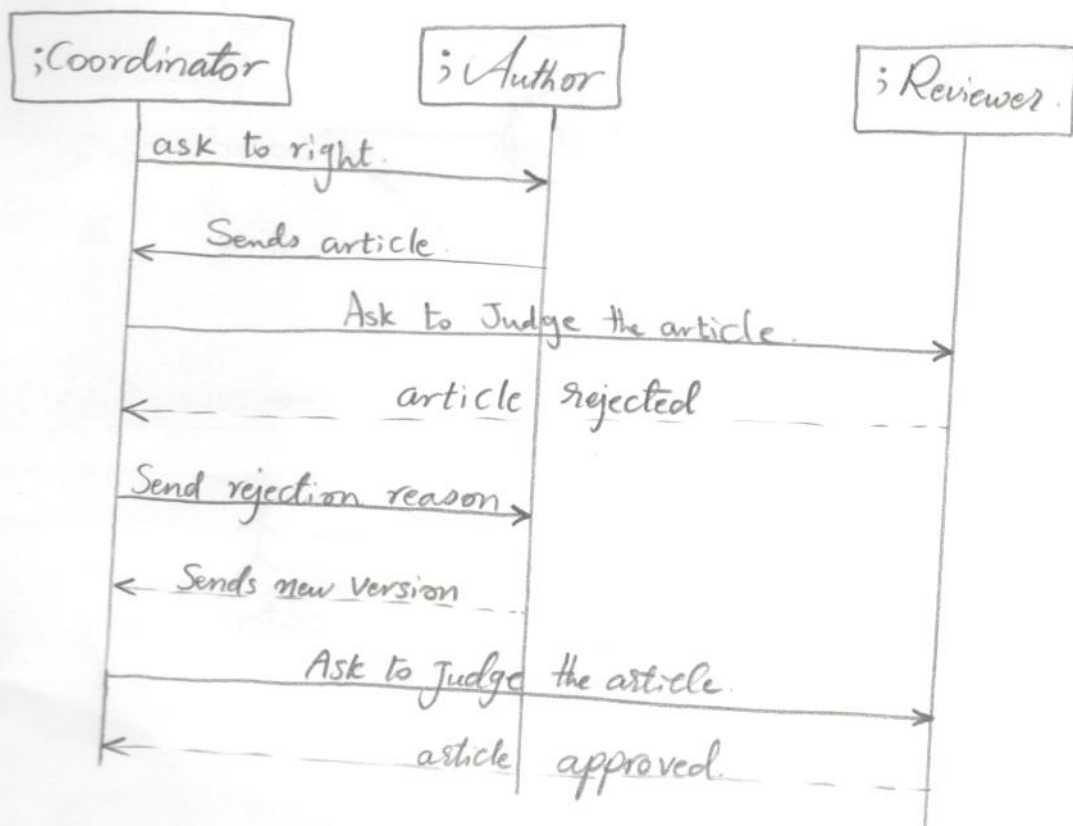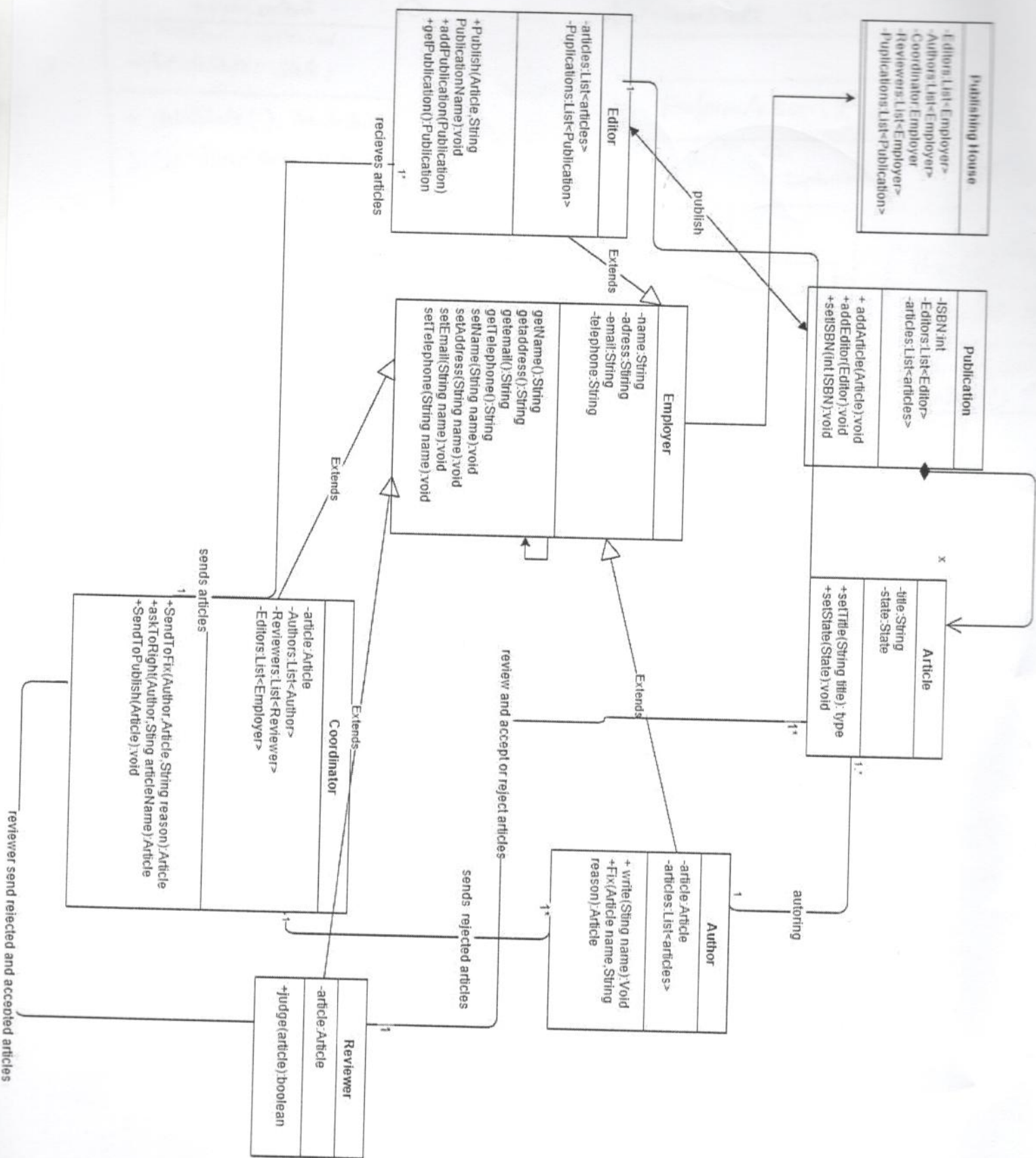
Question 2 :-

Contribute article

```
          ● (initial state)
          │
          ▼
   ┌──────────────────┐
   │ authoring State  │◀─────────────────┐
   └──────────────────┘                  │
          │                              │
          │ Submit article    article Rejected
          ▼                              │
   ┌──────────────────┐                  │
   │   under review   │──────────────────┘
   └──────────────────┘
          │
          │ article approved
          ▼
   ┌────────────────────────┐
   │  Ready for Publishing  │
   └────────────────────────┘
          │
          ▼
   ┌──────────────────┐
   │    Published     │
   └──────────────────┘
```

(annotation in blue): Here you miss the action event / action — Mohamed Saed

Question 3 :-

```
┌──────────────┐      ┌──────────┐          ┌─────────────┐
│ ;Coordinator │      │ ; Author │          │ ; Reviewer  │
└──────┬───────┘      └────┬─────┘          └──────┬──────┘
       │   ask to right    │                       │
       │──────────────────▶│                       │
       │   Sends article   │                       │
       │◀──────────────────│                       │
       │   Ask to Judge the article                │
       │──────────────────────────────────────────▶│
       │       article rejected                    │
       │◀──────────────────────────────────────────│
       │ Send rejection reason                     │
       │──────────────────▶│                       │
       │  Sends new Version│                       │
       │◀──────────────────│                       │
       │   Ask to Judge the article                │
       │──────────────────────────────────────────▶│
       │       article approved                    │
       │◀──────────────────────────────────────────│
```

UML Class Diagram

Publishing House
- Editors:List<Employer>
- Authors:List<Employer>
- Coordinator:Employer
- Reviewers:List<Employer>
- Publications:List<Publication>

Editor
- articles:List<articles>
- Puplications:List<Publication>

+Publish(Article,String PublicationName):void
+addPublication(Publication)
+getPublication():Publication

Publication
- ISBN:int
- Editors:List<Editor>
- articles:List<articles>

+ addArticle(Article):void
+ addEditor(Editor):void
+setISBN(int ISBN):void

Employer
- name:String
- adress:String
- email:String
- telephone:String

getName():String
getaddress():String
getemail():String
getTelephone():String
setName(String name):void
setAddress(String name):void
setEmail(String name):void
setTelephone(String name):void

Article
- title:String
- state:State

+setTitle(String title): type
+setState(State):void

Coordinator
- article:Article
- Authors:List<Author>
- Reviewers:List<Reviewer>
- Editors:List<Employer>

+SendToFix(Author,Article,String reason):Article
+askToRight(Author,String articleName):Article
+SendToPublish(Article):void

Author
- article:Article
- articles:List<articles>

+ write(String name):Void
+Fix(Article name,String reason):Article

Reviewer
- article:Article

+judge(article):boolean

Relationship labels: publish, Extends, recieves articles, sends articles, review and accept or reject articles, sends rejected articles, reviewer send rejected and accepted articles, autoring

Question 4 :-

```
┌─────────────────────────────┐                    ┌─────────────────────────────────────┐
│       ArticleContent        │◇───────────────→ 1 │ ArticleState   «Interface»          │
├─────────────────────────────┤                    ├─────────────────────────────────────┤
│ - boolean : approved ;      │                    │                                     │
│ - ArticleState : state ;    │                    ├─────────────────────────────────────┤
├─────────────────────────────┤                    │   PerformAction (); Void            │
│ + getState (); ArticleState │                    └─────────────────────────────────────┘
│ + SetState (ArticleState);  │                                    △
│             Void.           │                                    ┊ implements.
└─────────────────────────────┘                                    ┊
```

```
┌───────────────────────┐   ┌───────────────────────┐   ┌───────────────────────┐
│      Authoring        │   │    UnderReview        │   │      Rejected         │
├───────────────────────┤   ├───────────────────────┤   ├───────────────────────┤
│ - ArticleState next;  │   │ - ArticleState next;  │   │ - ArticleState nex    │
├───────────────────────┤   │ - boolean approved    │   ├───────────────────────┤
│ PerformAction ()      │   ├───────────────────────┤   │ performAction ()      │
│ getNext ()            │   │ perform Action ()     │   │ getNext ()            │
└───────────────────────┘   │ getNext ()            │   └───────────────────────┘
                            └───────────────────────┘
```

```
┌───────────────────────┐           ┌───────────────────────┐
│  ReadyForPublishing   │           │      Published        │
├───────────────────────┤           ├───────────────────────┤
│ - ArticleState next   │           │                       │
├───────────────────────┤           ├───────────────────────┤
│ PerformAction ();     │           │ performAction ().     │
│ getNext ();           │           │                       │
└───────────────────────┘           └───────────────────────┘
```

Question 6 :-

```
Public class Article Content () {

        Private  ArticleState state;
        Private  boolean      approved = false;
        Public Article Content () {

            State = new Authoring State (this);
        }

        Public Void Set State (Article State state) {
            this. State = State;
        }

        Public  Article State  getState () {
            Return  State;
        }

                    // end of Class
```

```
Public Interface  Article State {

        Public void  perform Action () ;

            // end of Interface
```

```
        Public void approve (boolean app) {
                approved = app;
        }

        Public boolean is Approved () {

            return approved ;
        }
```

```
public class Authoring State implements ArticleState {

    Private ArticleState  next ;

    Public Authorising State ( Article Content X) {

        next = new Under Review (X);
    }


    Public  Void perform Action() { }.

    Public ArticleState getNext() { return next; }.

// end of class.
```

```java
Public class UnderReview implements ArticleState {
    Private ArticleState next;
    Private boolean approved;

    Public UnderReview (Article Content content) {
        approved = content.isApproved();
    }
    Public void performAction () {
        if (approved)
            next = new ReadyForPublishing();
        else
            next = new Rejected ();
    }

    Public ArticleState getNext () {
        return next;
    }
}
```

```java
Public class Rejected implements ArticleState {
    Private Article State next;
    Public void PerformAction () {
        next = new Authoring ();
    }
    Public Article State getNext () {
        return next;
    }
}
```

```java
Public class Ready for Publishing implements ArticleState {
    Private ArticleState next;
    Public void Perform Action () {
        next = new Published;
    }
    Public Article State getNext () { return next; }
}
```
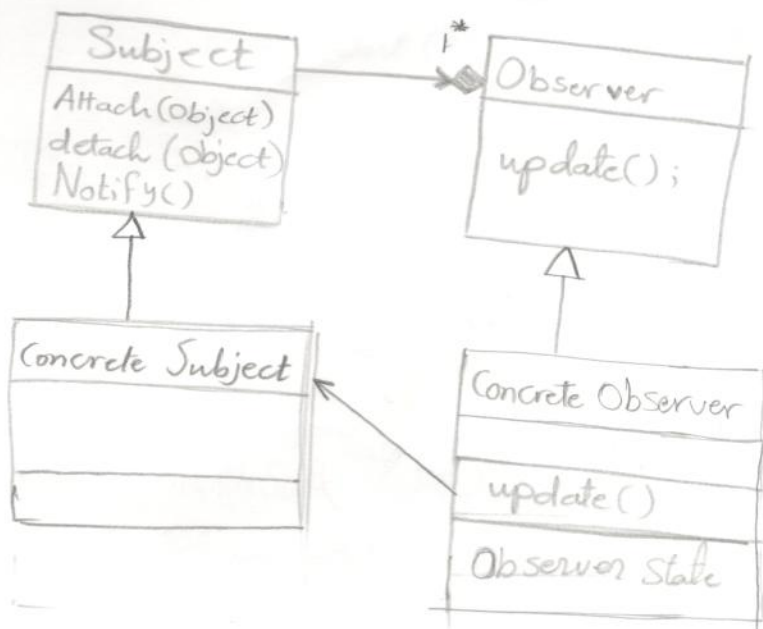
```java
Public class Published implements Article State {
    Public void perform Action () {}
}
```

- Question 7:-

@

Subject
| Attach (object) |
| detach (object) |
| Notify() |

Observer
| update(); |

Concrete Subject

Concrete Observer
| update () |
| Observer State |

ⓑ

Product

Creator
| Create() |

Concrete Product

Concrete Creator
| Creat() |