Please write your name **only** in the **last page**.

Fall 2010 Final Exam Sample Solutions

CS 319 Object-Oriented Software Engineering

Instructors: Uğur Doğrusöz and Bedir Tekinerdoğan

Reminders

- *Time: 120 minutes (2 Hours)*
- *Write your name and sign only in the last page as indicated.*
- *Show your work and reasoning clearly and write legibly, only within the space provided for each question. Do not detach any page(s).*
- *For diagrams and programs, clarity is important; if your diagrams or programs are sloppy and hard to read, you will lose points.*
- *From the time you receive your exam script, you will have 60 minutes to read all questions and make sure you understand what is expected from you. During this time you may ask your instructor any questions should you require any clarification. After that you may not seek any type of feedback from the instructor.*

Q1	*15*	
Q2	*20*	
Q3 *a-c*	*50*	
Q3 *d*	*15*	
Total	***100***	

Question 1: Fill in the blanks [15 pts]

Fill in the blanks below, with a word or more per blank, to form valid statements. When provided with options, choose one and cross others.

a) Whereas scenarios are concrete examples illustrating a single case, use cases are abstractions describing all possible cases.

b) What makes software engineering such a difficult and challenging discipline today is complexity and change. Whereas, in the old days, software specifications could be easily drafted and finalized, and entire development could be performed by a single programmer.

c) The analysis model is composed of three individual models: the functional model, represented by use cases and scenarios, the analysis object model, represented by class and object diagrams, and the dynamic model, represented by state machine and sequence diagrams.

d) In analysis, sequence diagrams are used to help identify new participating objects and/or existing ones' missing behavior.

e) In object-oriented analysis, developers build a model describing the application domain.

f) During system design, developers identify and prioritize design goals (i.e., qualities of the system that they should optimize).

g) If two subsystems are loosely coupled, they are relatively independent, so modifications to one of the subsystems will have little impact on the other.

h) The set of operations of a subsystem that are available to other subsystems form the subsystem interface.

i) A peer-to-peer architectural style is a generalization of the client/server architectural style in which subsystems can act both as client and as servers, in the sense that each subsystem can request and provide services.

j) During object design, we close the gap between the application objects and the off-the-shelf components by identifying additional solution objects and refining existing objects. Main object design activities include reuse / service specification and object model restructuring / object model optimization.

k) Last but not least, my name is written only on the last page of this exam paper as instructed.

Question 2: OO Design and Implementation [20 pts]

Suppose we have the four classes below which implement an indirect doubly-linked list mechanism for `Card` objects and use it in a Java application. Some details are left out for brevity. Please study the implementation of these classes and refer to them for the rest of this question.

```
class TestQuestion {
    public static void main(String [] args)
    {
        CardList myCardList = new CardList();
        Card newCard;

        for (int i = 1; i <= 2; i++)
        {
            int j = i + 5;
            newCard = new Card(i, j);
            myCardList.insertCard(newCard);
        }
    }
}
class Card {
    public Card(int s, int r) { suit = s; rank = r; }
    public int suit;
    public int rank;
}
class ListElement {
    public ListElement prev;
    public ListElement next;
    public Card card;
}
class CardList {
    // This method creates a new list element on the heap associated with
    // the input card and inserts into the beginning of this list.
    public void insertCard(Card card) { ... }

    // This method removes the input card from this list.
    public Card removeCard(Card card) { ... }

    // This method returns the first card of this list.
    public Card getFirst() { ... }

    ...

    public int noOfElements;
    public ListElement first;
    public ListElement last;

}
```

(a) *[2 pts]* Briefly discuss if the implementations of any of the classes above violate any one of the basic principles of Object-Oriented programming. Assume that any such problems have been fixed for the remainder of this question.

The basic principle of information hiding / data encapsulation has been violated in these classes since instance variables are made public. Appropriate getter/setter methods should have been used to properly access and manipulate such instance data. We will assume all such variables have been defined "protected" for the rest of the answers.

(b) *[4 pts]* Implement a new class `RestCardList` (reusing any of the classes above when possible) and behaves exactly the same as `CardList` except that the number of elements a `RestCardList` takes is limited by an integer parameter `maxNoOfCards`. This parameter is to be passed into its constructor during instantiation.

```
class RestCardList extends CardList {
    private int maxNoOfCards;

    public RestCardList(int m) { maxNoOfCards = m; }

    public void insertCard(Card newCard) {
        if (noOfElements < maxNoOfCards)
            super.insertCard(newCard);
    }
}
```

(c) *[8 pts]* Implement a new class `CardStack` (reusing any of the classes above when possible) and maintains a stack (a last-in first-out data structure) of cards. Make sure your class does not allow insertion of any objects other than cards into the stack.

// Use the **Adapter** design pattern and compose a `CardList` into `CardStack`

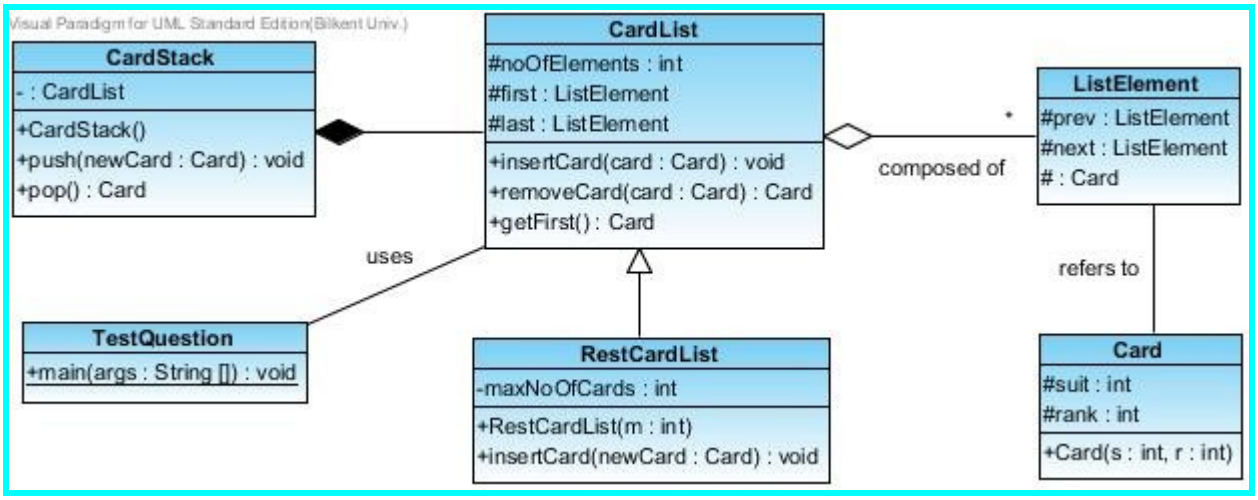
```
class CardStack {
    private CardList theCards;

    public CardStack() { theCards = new CardList(); }

    public void push(Card newCard) {
        theCards.insertCard(newCard);
    }

    public Card pop() {
        return theCards.removeCard(theCards.getFirst());
    }
}
```

(d) *[6 pts]* Suppose all the code provided above and requested from you in parts (b) and (c) are collected in a Java package. Reverse engineer the code in this package to produce an Object Design Model in UML.



Question 3: OOA/D and UML [65 pts]

Consider the following problem description:

You have been hired by Southeastern North Anatolia Polytechnic University (SNAPU) to design a new digital library system for research publications. The system will allow users to download and view information about published research papers and technical reports. The system will also track information about where papers were published, and which papers are cited or referenced by any particular paper.

The SNDPU Provost's office has identified the following features for development:

- The system will be implemented in Java as a desktop application running on an ordinary PC.
- Because the system can only be used by the faculty, students and staff of SNAPU, users entering the system should be prompted for their university-wide userID and password, which can be maintained by a user management library developed earlier by the University.
- Users should be able to display a list of all publications for a particular author. The system stores data about conference papers, university technical reports and journal articles. Naturally, the title, the list of author(s), and the year of publication are tracked for each paper. Additional information varies by the type of publication: conference papers include conference name and location; journal articles include journal name, publication month, and publisher; and technical reports have unique IDs assigned by the University.
- Users should be able to display a list of all papers in the proceedings of a specified conference or a list of all articles in a specified edition of a journal. Note that proceedings of a conference contain all papers that were presented in that conference held in a certain location during a certain time period. Similarly, journals are organized into editions, where each edition made available at a certain date consists of a number of papers.
- Users should be able to search the library for all publications containing a specified list of keywords in their content. Luckily there is a couple of open-source, free software libraries implementing such efficient search methods, one using domain indexing, the other using function-based indexing. The actual library tool to use should be able to be determined by the system administrator dynamically depending on the current library content and indexing mechanisms provided by the database management system in use.

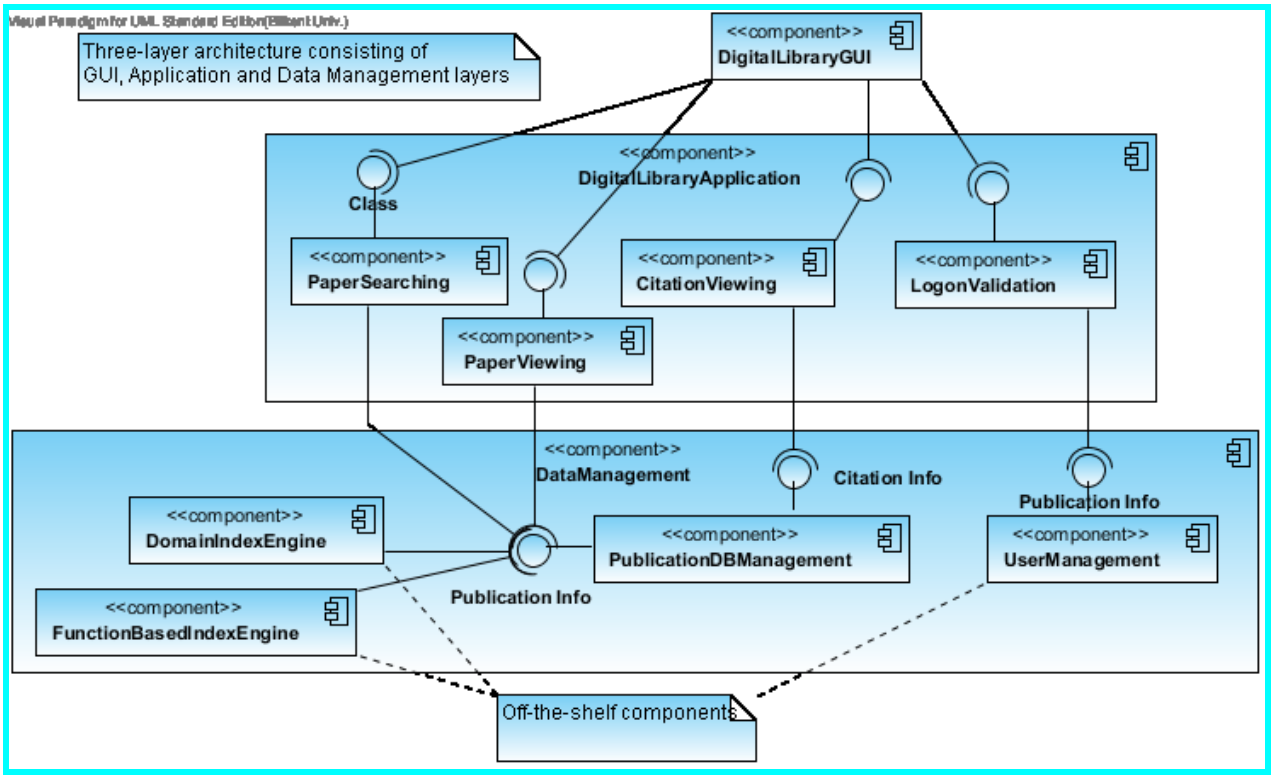
- Users should be able to download publications stored in the system. The system should keep track of the number of times a publication is downloaded. The manuscript of a publication often contains large figures.

- Users should be able to display a list of all papers which cite/reference a particular publication. It is common in a scientific publication to refer to other, earlier publications as related work or background material. In addition, a particular user should be able to register for a particular publication for any citations. This way, a person could get notifications by email upon addition of a new publication to the library citing their work.

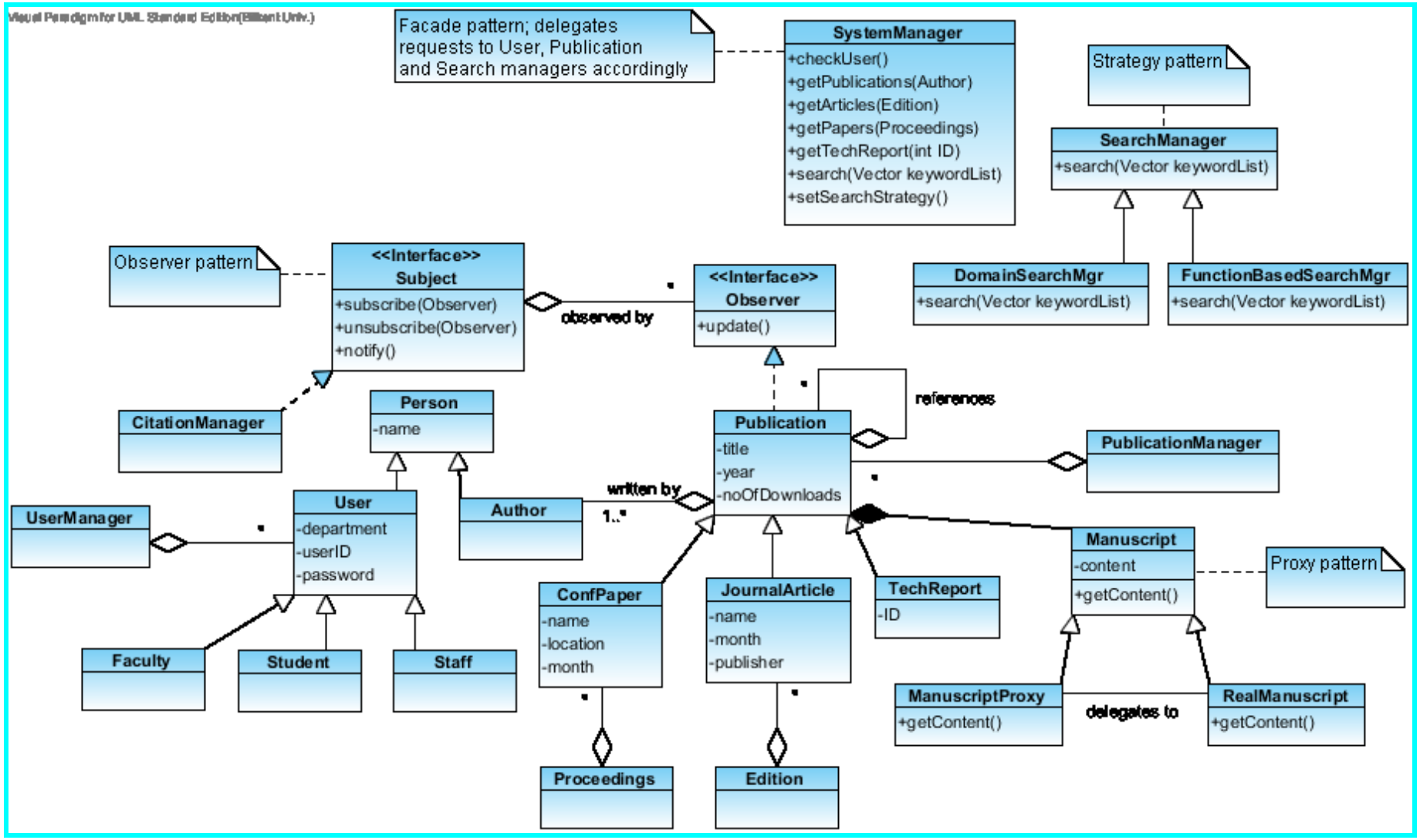
If anything is unclear, please state your assumptions when you answer the following questions.

(a) *[15 pts]* Perform a quick system design for the above system and express your findings with a UML diagram. Specify any architectural style and any layering used for your design. Also specify any off-the-shelf components used.

Below is a design using the three-layer architectural style; other architectural styles such as Client/Server or MVC could also be used.



(b) *[30 pts]* Show the UML class diagram for the digital library system such that it can support implementation of all the features specified above except user interface and data management/persistence related classes. Be sure to include all the details (instance variables, methods, and relations between classes) necessary to implement the main functional requirements. You do not need to include getter/setter methods, specify visibility, or operation contracts in the class diagram. Make sure to use and specify at least two distinct design patterns as appropriate.


Facade pattern; delegates requests to User, Publication and Search managers accordingly
SystemManager
+checkUser()
+getPublications(Author)
+getArticles(Edition)
+getPapers(Proceedings)
+getTechReport(int ID)
+search(Vector keywordList)
+setSearchStrategy()
Strategy pattern
SearchManager
+search(Vector keywordList)
DomainSearchMgr
+search(Vector keywordList)
FunctionBasedSearchMgr
+search(Vector keywordList)
Observer pattern
<<Interface>>
Subject
+subscribe(Observer)
+unsubscribe(Observer)
+notify()
observed by
<<Interface>>
Observer
+update()
CitationManager
Person
-name
Publication
-title
-year
-noOfDownloads
references
PublicationManager
UserManager
User
-department
-userID
-password
Author
written by
1..*
Manuscript
-content
+getContent()
Proxy pattern
ConfPaper
-name
-location
-month
JournalArticle
-name
-month
-publisher
TechReport
-ID
Faculty
Student
Staff
ManuscriptProxy
+getContent()
delegates to
RealManuscript
+getContent()
Proceedings
Edition

(c) *[5 pts]* Suppose the University is interested in making the system available in C# (.NET environment) as well in later versions of the software. Briefly discuss how you would design the user interface component of your software, taking this multi-platform support as a design goal into account. Also mention whether or not any remodeling of the proposed design (part (b)) would be necessary to make porting less painful.

Application logic component designed earlier in part (b) would not need to be remodeled since there is no platform-specific logic/code in this subsystem.

When designing the user interface component of the system, however, one should try to separate platform-specific graphics code from the rest of the code using design patterns such as Bridge, Command, and Abstract Factory as appropriate.

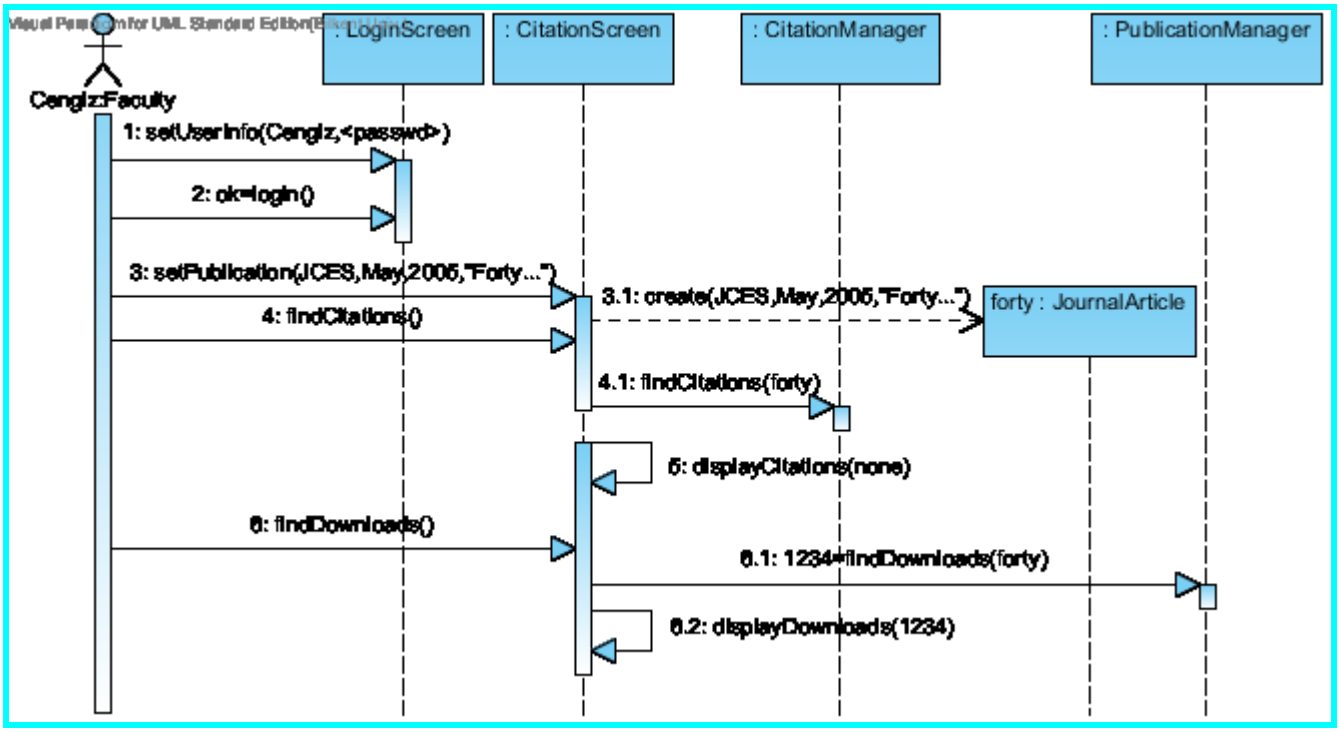
(d) *[15 pts]* Consider the following scenario:

> Professor Cengiz wants to better understand the impact of his research in his field. As part of his efforts to do so, he logs on to use the digital library system and searches for all papers citing his paper entitled "Forty nine ways to guarantee success in CS319" that appeared in May 2005 issue of Journal of Computer Engineering Survival (JCES). It turns out that his paper has received no citations. However, he finds out that his article has been very popular among the students, having been downloaded a total of 1234 times since it was added to the digital library system.

Name the use case of the digital library system that would cover this scenario. Then construct a sequence diagram that describes this particular scenario in terms of the interaction between software/solution objects as designed earlier in part (b). Feel free to add user interface objects if needed.

"Check Citations" would be an appropriate name for the use case covering this scenario.

Please write your name **only** in the **last page**.



Mini Dictionary:

Brevity	Bir fikrin kısaca ifade edilmesi
To cite	Atıfta bulunmak
Content	İçerik
To draft	Taslağını yapmak
Edition	Baskı, edisyon
Gap	Boşluk, aralık
To hire	İşe almak
Impact	Etki
Manuscript	Bir eserin metni
To port	Taşımak
Sloppy	Yarım yamalak, baştan savma
To track	İzini takip etmek
To violate	İhlal etmek

I hereby affirm that the work submitted in this examination is my own exclusively.

Name & Signature: Uğur Doğrusöz Bedir Tekinerdoğan **Section:**