



Al Imam Mohammad Ibn Saud Islamic University
College of Computer and Information Sciences
Computer Science Department

Course Title:	Software Engineering
Course Code:	CS310
Course Instructors:	Dr Alqahtani, Ms Aljaloud
Exam:	Final Exam
Semester:	Spring 2019
Date:	27/08/1440 – 03/05/2019
Duration:	120 minutes
Marks:	40

Privileges: ☐ Open Book ☐ Open Notes
☐ Calculator Permitted ☐ Laptop Permitted

Student Name:	
Student ID:	MODEL ASNWS
Section No.:	

Instructions:

1. Answer 4 questions; there are 4 questions in 13 pages.
2. Write your answers directly on the question sheets. Use the ends of the question pages for rough work or if you need extra space for your answer.
3. If information appears to be missing from a question, make a reasonable assumption, state your assumption, and proceed.
4. **No** questions will be answered by the invigilator(s) during the exam period.

Official Use Only		
Question	Student Marks	Question Marks
1		10
2		11
3		9
4		10
Total		40



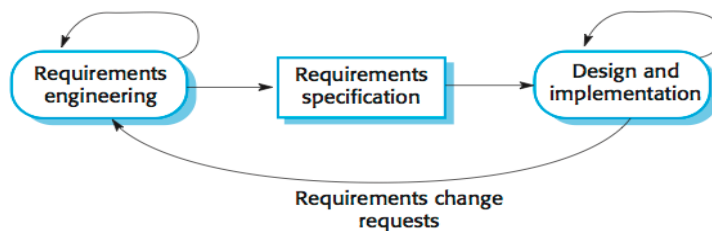
Question 1:

To be answered in (25) Minutes [] / 10 Marks

(0.5 points each) Multiple choices. Choose ONLY the best answer.

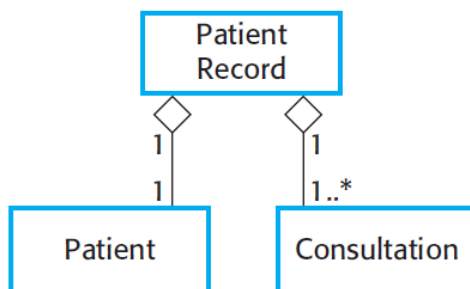
1. Dissatisfaction with the overheads involved in software design methods of the 1980s and 1990s led to the creation of agile methods. One of these methods is:
 - a. Focus on the code rather than the design
 - b. Focus on the design rather than the code
 - c. Focus on the requirement engineering rather than the design
 - d. None of the above.

2. The following diagram is representing



- a. Agile development
 - b. Plan-based development
 - c. Plano-based development: spiral model.
 - d. Agile development: scrum model.
3. Which list represents the methods that introduce agility:
 - a. Scrum, eXtreme Prgraomming, waterfall, Spiral Models, Crystal.
 - b. Both a and b.
 - c. Scrum, eXtreme Programming only.
 - d. Scrum, eXtreme Programming, Rational Unified Process, Kanban, Crystal.
4. The difference between functional and non-functional requirements is:
 - a. A functional requirement describes *how* a software system should do, while non-functional requirements place constraints on *what* the system will do so.
 - b. Functional requirements apply to the system as a whole and non-functional apply to the system as individuals.
 - c. A functional requirement describes *what* a software system should do, while non-functional requirements place constraints on *how* the system will do so.
 - d. a and c.
5. The difference between user requirements and system requirements is:
 - a. The system requirements are written for customers.
 - b. The system requirements are defining what should be implemented.
 - c. The user requirements are written for customer.
 - d. None of the above.

6. Software requirements are written as natural language sentences supplemented by diagrams and tables. This introduces challenges and problems to the software requirements documents such as lack of clarity, confusion and requirements amalgamation. The proposed solutions for this is:
 - a. Design and implementation documents.
 - b. Form based and Tabular specification.
 - c. Tabular specification only.
 - d. All of the above.
7. Use case diagram and sequence diagram are types of
 - a. Interaction models.
 - b. Context models.
 - c. Structural models.
 - d. Behavioural models.
8. The following diagram represents



- a. The aggregation association.
 - b. Generalization hierarchy.
 - c. Activity model.
 - d. State diagram.
9. _____ are used to illustrate the operational context of a system – they show what lies outside the system boundaries.
 - a. Interaction models.
 - b. Context models.
 - c. Structural models.
 - d. Behavioural models.
 10. In _____, all tests are rerun every time a change is made to the program to ensure that there is no errors have been introduced by this change.
 - a. integration testing.
 - b. regression testing.
 - c. unit testing.
 - d. system testing.



11. ____ is one of the user testing, where is it test the software at the developer's site:
- Alpha testing.
 - User testing.
 - Beta testing.
 - a and c.
12. Identifying ____ is often a difficult part of object oriented design.
- Object class.
 - Object methods.
 - Object attributes.
 - All of the above.
13. The difference between static and dynamic models is that static model
- Describe the static structure of the system in terms of object classes and relation.
 - Describe the dynamic interactions between objects.
 - Both a and b.
 - None of the above.
14. A ____ is a way of reusing abstract knowledge about a problem and its solution.
- design pattern.
 - class diagram.
 - sequence diagram.
 - interaction model
15. ____ development is an approach to software development in which the source code of software system is published and volunteers are invited to participate in the development process.
- Open source.
 - Close source.
 - Test-driven.
 - Model-driven.
16. GPL stands for
- The GNU General Public License.
 - The GNU Lesser General Public License.
 - The Berkley Standard Distribution.
 - Bot a and b.
17. Unit testing may be created and implemented for:
- A method
 - An object class
 - A component
 - All of the above



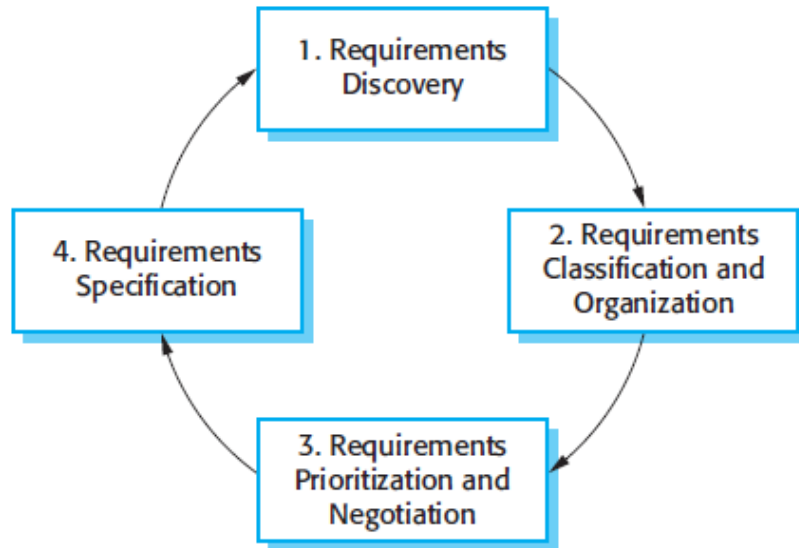
-
18. ___ a test that can find design and coding errors:
- a. White-box testing
 - b. Black-box testing
 - c. Functional testing
 - d. b and c
19. Model transformation is
- a. the process of transforming from system model to another.
 - b. the process of transforming from system model to program.
 - c. the process of transforming from program code to another.
 - d. the process of transforming from program code to system model.
20. Implementing the object design model in a programming language is the goal of ____
- a. forward engineering.
 - b. reverse engineering.
 - c. model transformation
 - d. refactoring.
-

Question 2:

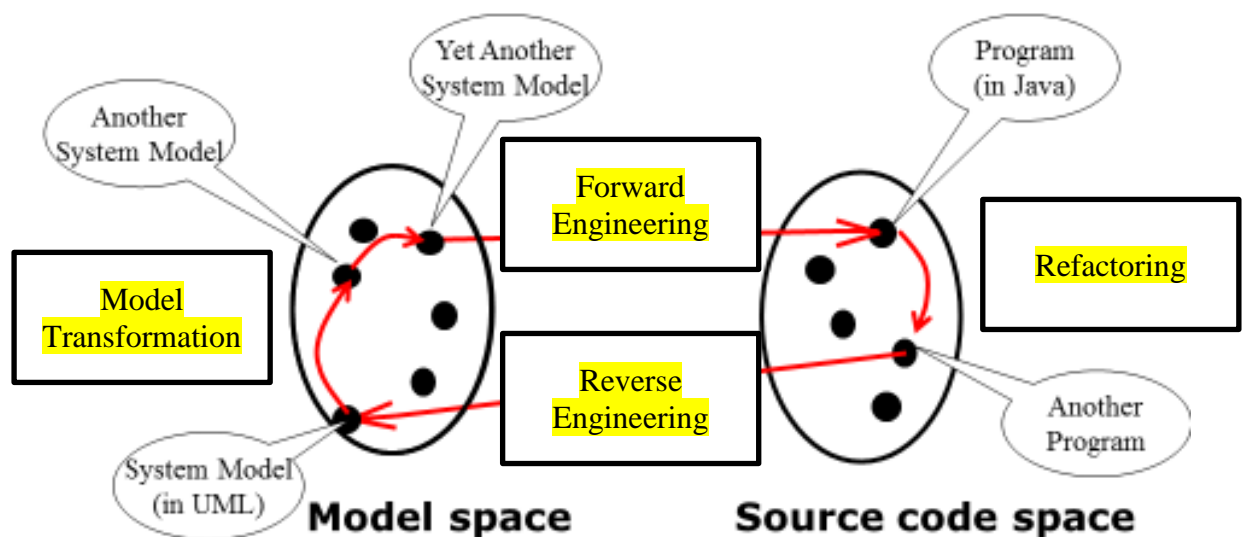
To be answered in (30) Minutes [] / 11 Marks

Short answer

1. (2 points) The requirement elicitation and analysis process consist of four stages: (fill the blank) Chapter 4 slide 43



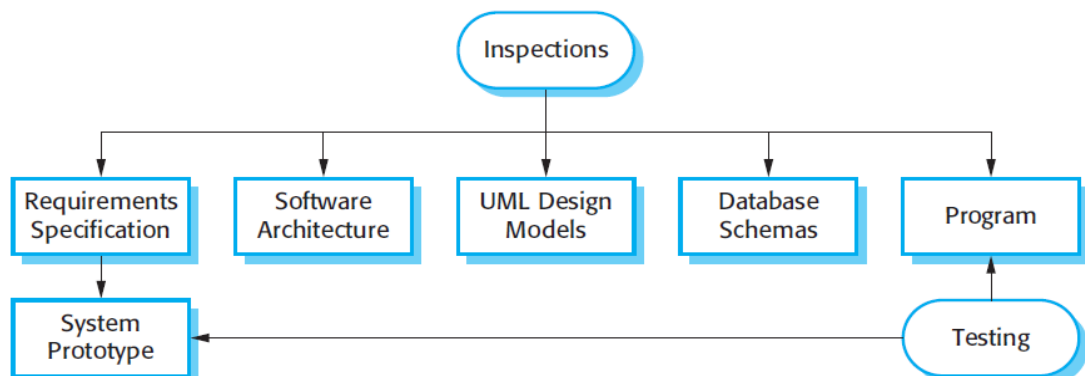
2. (2 points) Four different types of transformations occur between model space and code space in software engineering. Name each transformation type in the following figure:



5. (1.5 points) What are the main three issues of software implementation that are not covered in programming texts:

-----Reuse, Configuration management, and Host-target development (Chapter 7 slide 34)-----

6. (1.5 points) Inspections and testing are complementary and not opposing verification techniques. Both should be used during the V & V process. From the following figure, does testing and inspection components posted on the correct place? If yes/no, please justify your answer.



Chapter 08 slide 10 and 11



7. **(2 points)** In software testing, V&V refer to Verification and Validation. What is the main difference between these two concepts? (Chapter 8 slide 8)

Verification: The software should conform to its specification

Validation: The software should do what the user really requires

8. **(2 points)** List two main differences between white-box testing and black-box testing

White-box testing	Black-box testing
1. Test structure	1. Test function
2. Find design and code errors	2. Can find requirements specification errors



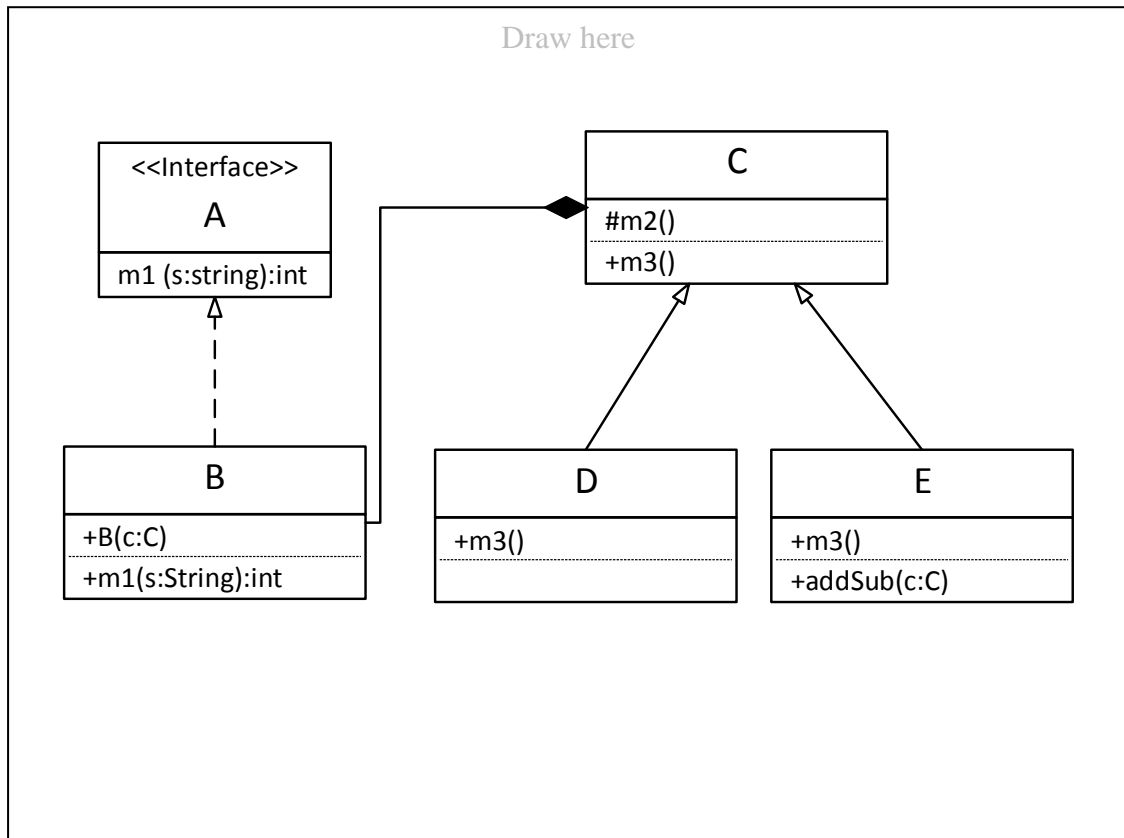
Question 3:

To be answered in (30) Minutes [] / 09 Marks

1. (4 points) Consider the code below. Draw the corresponding UML Class Diagram.

```
public interface A {  
    int m1( String s );  
}  
  
public class B implements A {  
    private C myC;  
    public B( C c ) {  
        myC = c;  
        myC.m3();  
    }  
    public int m1(String s){  
        return 3;  
    }  
}  
  
public class C {  
    protected void m2(){  
        System.out.println("hi");  
    }  
    public void m3();  
}
```

```
public class D extends C {  
    public void m3() {  
        System.out.println("yes");  
    }  
}  
  
public class E extends C {  
    public void m3() {  
        for( C c : subs ) {  
            c.m2();  
        }  
    }  
    public void addSub(C c ) { subs.add(c);  
    }  
}
```

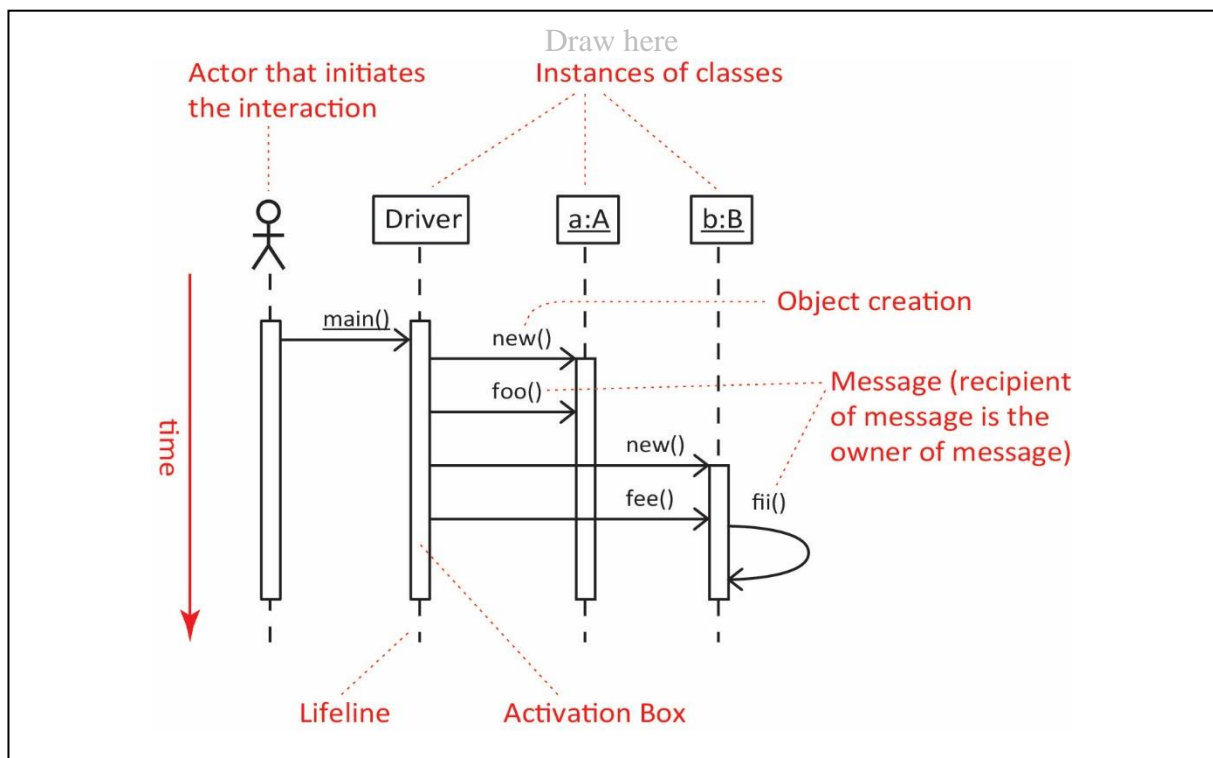


2. (3 points) Consider this code below. Draw a sequence diagram for the case when the system is run.

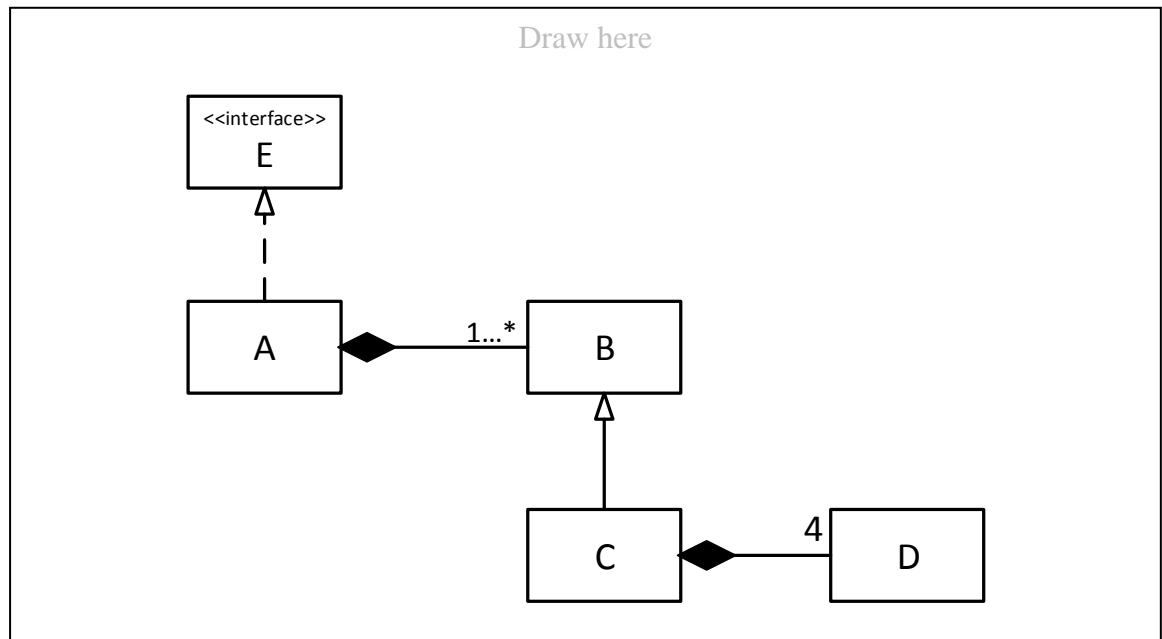
```
public class Driver {
    public static void main(String[] args)
    {
        A a = new A();
        a.foo();
        B b = new B();
        b.fee();
    }
}
```

```
public class A {
    public A() {}
    public void foo() {}
}

public class B {
    public B() {}
    public void fee() {
        fii();
    }
    public void fii() {}
}
```



3. (2 points) Draw a UML Class Diagram that represents the following situation. There is a class A with an ArrayList of B objects. Class B is abstract with subclasses C and D. Class C is composed of four D object. Also, class A is composed of 1 or more B objects, and implements interface E.



Question 4:

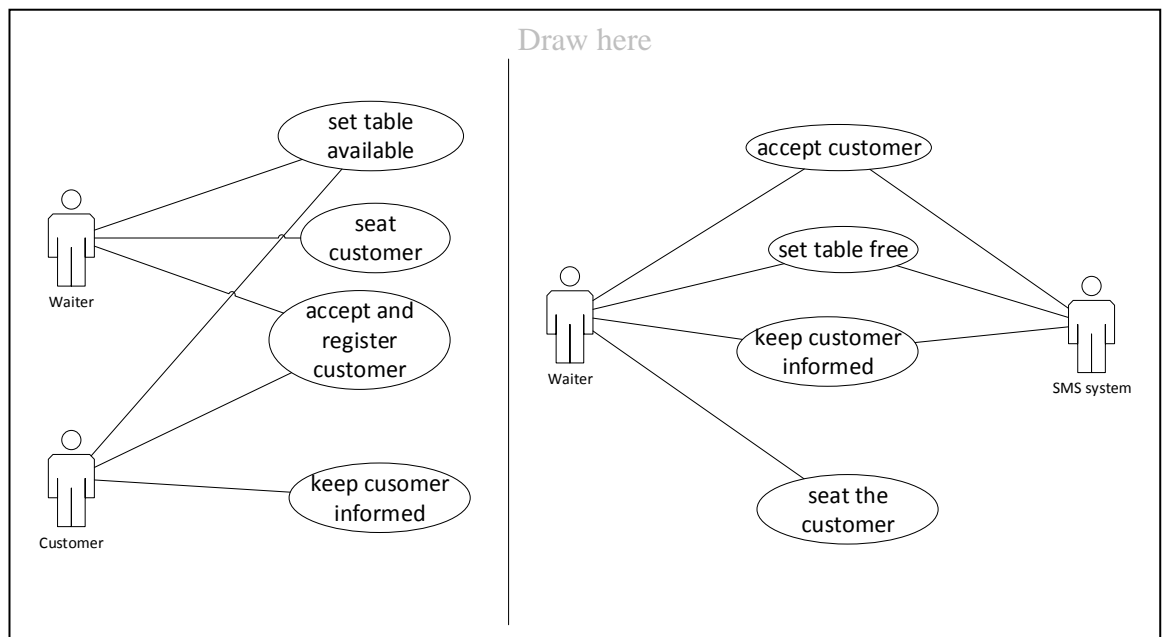
To be answered in (35) Minutes [] / 10 Marks

1. (3 points) Restaurant queue management

At a popular restaurant within a mall, a system is used to manage the waiting queue. When a new customer arrives, to reserve a table, the waiter registers the customer name and mobile phone number, the system then sends a confirmation SMS to the customer's phone. At this point the customer has a table reserved. Then the customer is free to walk away and e.g. visit the shops in the mall. Every five minutes the system will send an SMS with the estimated residual waiting time. When the table for the customer gets available, a waiter sets the table as ready and the system sends an SMS to the customer asking him to get to the restaurant as soon as possible to get the reserved table. As the customer gets back to the restaurant, the waiter seats the customer at the reserved table. If the customer does not show up within a fixed time, then the next customer in the waiting queue is recalled.

- a- Draw a UML use case diagram showing the system context with the main actors (≤ 2 actors) and the relative use cases (≤ 5 use cases).

Hint: two alternative models are possible, depending on whether the SMS system is within (left hand side diagram) or outside (right hand side diagram) the system boundary.



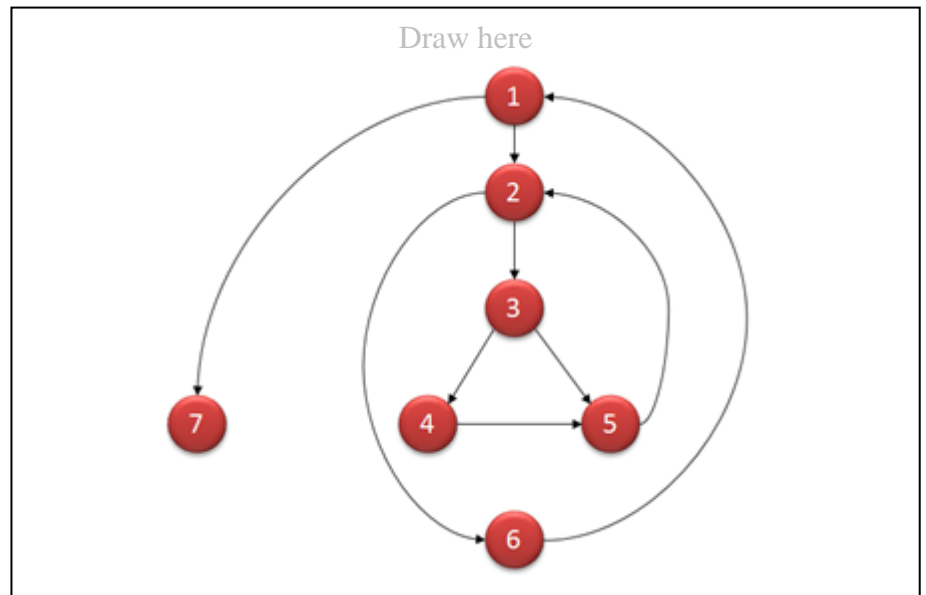
2. (7 points) Consider the code below. (a) Draw a corresponding flow graph.

```

i = 0;
n=4;

while (i<n-1) do
    j = i + 1;
    while (j<n) do
        if A[i]<A[j] then
            swap(A[i], A[j]);
        end do;
        i=i+1;
    end do;
end do;

```



(b) Determine the cyclomatic complexity of the resultant control flow graph.

Method 1: $V(G) = 9 - 7 + 2 = 4$

Method 2: $V(G) = 3 + 1 = 4$ (Condition nodes are 1, 2 and 3 nodes)

Method 3: Regions = 4

(c) Find basic independent path set.

Path 1: 1, 7

Path 2: 1, 2, 6, 1, 7

Path 3: 1, 2, 3, 4, 5, 2, 6, 1, 7

Path 4: 1, 2, 3, 5, 2, 6, 1, 7