

School of Engineering and Applied Science

**CS2020 Software Engineering
Second Year Examination**

CLOSED BOOK

**Date: 21st January 2014
Time: 09:30 – 12:30
Duration: 3 hours**

Instructions to Candidates

- 1. Answer ALL questions from Section A (40 marks)**
- 2. Section A contains EIGHT questions**
- 3. Answer THREE questions from Section B (60 marks)**
- 4. Section B contains FOUR questions, each question is worth 20 marks.**
- 5. Use of calculators is NOT allowed**

Materials provided

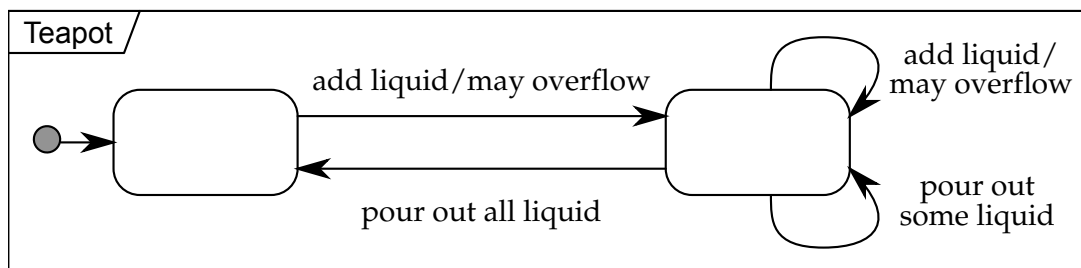
- 1. Answer booklets**

This exam paper cannot be removed from the exam room

Section A — Answer ALL questions

1. STATE TWO desirable **emergent** properties of a home intruder alarm system. Moreover, one of the properties must be **functional** and the other one **non-functional**. (4 marks)

2. Consider the following UML diagram:



- a) STATE the type of the above UML diagram. (2 marks)
- b) SUGGEST helpful names to write inside the two empty boxes. (2 marks)
- c) DESCRIBE how to extend the diagram so that it also describes what happens when the teapot is dropped on the floor. (4 marks)
3. GIVE TWO reasons why **domain** or **business analysis** is often critical to the success of a project. (4 marks)

4. **REWRITE** the following requirement to make it more precise and thus easier to verify:

The system should be easy to use by administrative staff and should be organised in such a way that user errors are minimised.

(4 marks)

5. **BRIEFLY EXPLAIN** the main difference between software architecture and framework. (4 marks)

6. The 'Gang Of Four' classified design patterns into three generic groups: **creational**, **structural** and **behavioural**.

- a) **STATE** the key design problem that the **behavioural** design patterns address. (2 marks)

- b) **DESCRIBE**, with the aid of a UML class diagram, the structure of the behavioural pattern known as the **State pattern**. (6 marks)

7. Assume a class model has been designed that features **collections**, **inner classes** and **inheritance**. **EXPLAIN** how to map such a class model to tables in a database. (6 marks)

8. **BRIEFLY EXPLAIN ONE** benefit of **Unit Testing**. (2 marks)

END OF SECTION A

Section B — Answer THREE questions

9. Assume that the project outlined below is managed using the Unified Process:

The system can be installed on a junction where two straight roads cross. It provides red-amber-green traffic signals for all traffic approaching the junction. An operator can switch the system off or on, as well as into manual or automatic mode.

When in automatic mode, the system detects traffic using sensors and sets the signals to let traffic through in a safe as well as fairly smooth and fair manner. When in manual mode, the operator can switch the direction of traffic permitted through the junction. Moreover, the system gives the operator a warning when a vehicle has been waiting for a green signal for over 2 minutes.

Further, assume that the project has been delivered in 11 iterations whose main objectives and deliverables are summarised in the following table:

main objectives	deliverables
1 get sample drivers and operators	collaboration contracts
2 identify and classify risks	elaboration plan
3 identify typical scenarios	detailed scenario descriptions
4 signal and sensor control	prototype that switches light when car detected
5 realistic simulation	junction simulator program
6 safe and reasonably efficient automatic mode	prototype of automation component
7 manual mode	working manual mode
8 safety in manual mode	safely working manual mode
9 safe automatic switching	cars through safely without operator
10 minimise driver waiting	cars through safely and efficiently
11 documentation	installation and operator manuals

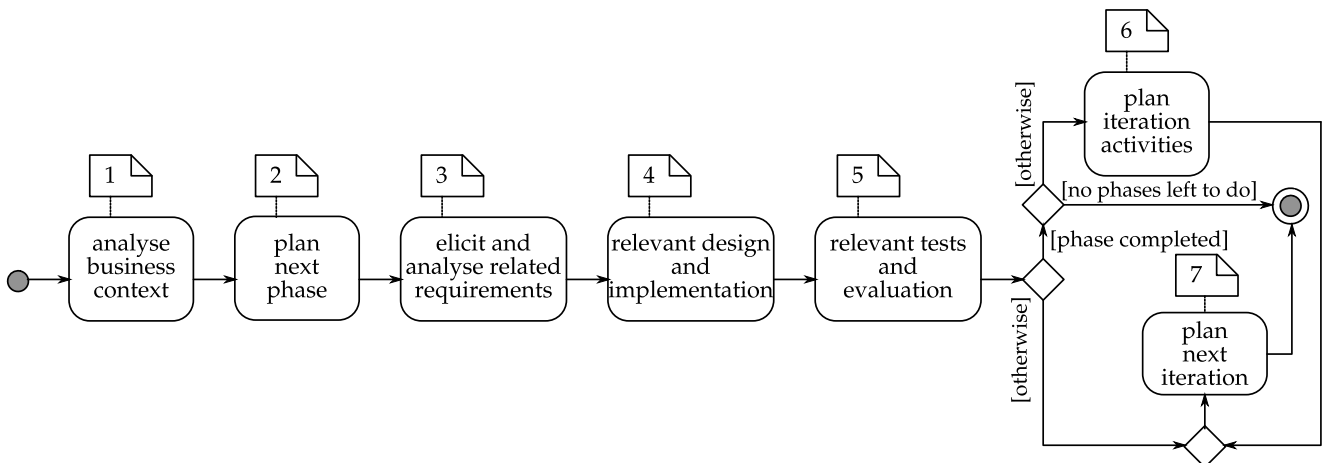
(question continues on next page. . .)

(Question 9 continued...)

a) NAME and briefly DESCRIBE the goals of EACH of the four Unified Process **phases**. (8 marks)

b) INDICATE which of the above iterations belong to which of the four Unified Process **phases**. (4 marks)

c) The following activity diagram is meant to illustrate some aspects of one Unified Process iteration but the activities are mixed up:

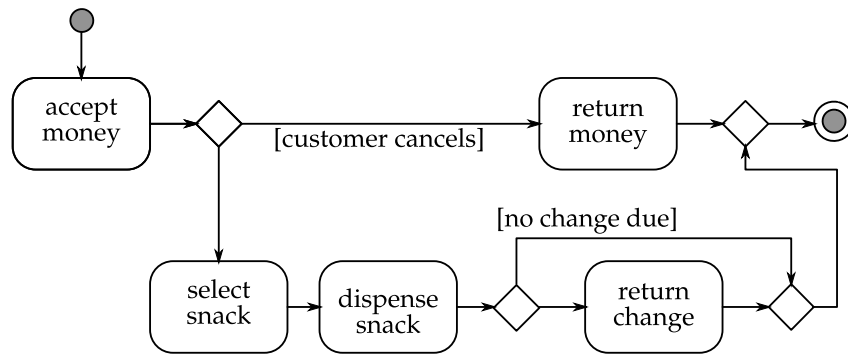


Using the numbers attached to the activities, STATE the correct order of these seven activities. (4 marks)

d) STATE TWO advantages and TWO disadvantages of using the **waterfall** approach instead of an **iterative** process when developing the road junction system described earlier.

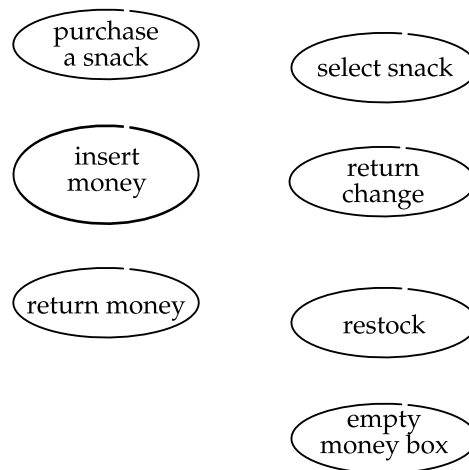
(4 marks)

10. Consider a certain type of vending machine that facilitates, among others, the process described by the following **activity diagram**:



Assume the manufacturer of the vending machine commissioned you to develop a computer system to control the machine.

The following is an incomplete **use case diagram** of the machine:



- DRAW** the above **use case diagram** and **ADD** actors into it, appropriately connecting them with use cases. (2 marks)
- DRAW** any appropriate relationships between use cases in your copy of the above use case diagram. (3 marks)

(question continues on next page. . .)

(Question 10 continued...)

c) Consider the following partially filled use case description form:

Use case number: 1	Name: Purchase a snack
Goal: Let customer purchase a snack from the vending machine.	
Brief description: Involves accepting money, selecting a snack and dispensing the snack+change.	
Actors: <i>(deliberately omitted)</i>	
Frequency of execution:	
Scalability:	
Criticality:	
Preconditions:	
Postconditions:	
Primary path:	
Alternatives:	
Exceptions:	

In the following tasks you will complete some of the items missing in this form.

- i) **STATE** the **non-functional** requirements missing in the above use case description form.

In your answer, you can make any reasonable assumptions on the details of the vending machine parameters and users' preferences.

(3 marks)

- ii) **STATE ONE precondition** and **ONE postcondition** of the use case.

(4 marks)

- iii) **DESCRIBE** the **primary path** of this use case.

(3 marks)

- iv) **DESCRIBE ONE alternative** and **ONE exception** of this use case.

(5 marks)

11.a) Consider a Java application which contains the following outline Java code with FIVE classes and an interface.

```
package pets;
public interface Trainable {
    public void beTrained();
}
```

```
package pets;
import ward.*;
public class Pet {
    private static int nextPetID = 1;
    private Owner owner;
    public Owner getOwner() { return owner; }
}
```

```
package pets;
public class Dog extends Pet implements Trainable {
    private String bark;
    private int trainingLevel;
    public void beTrained() { ... }
    public String getBark() { return bark; }
    public int getTrainingLevel() { return trainingLevel; }
    public void setTrainingLevel(int level) { trainingLevel = level; }
}
```

```
package pets;
public class CyberPet extends Pet implements Trainable {
    private int remainingLives;
    private int trainingLevel;

    public void beTrained() { ... }
    public int getTrainingLevel() { return trainingLevel; }
    public void setTrainingLevel(int level) { trainingLevel = level; }
    public void reduceLive() { remainingLives--; }
}
```

```
package pets;
public class Fish extends Pet {
    // detail omitted
}
```

```
package ward;
import pets.*;
public class Owner {
    private Pet[] myPets;

    /** Constructor: Create a new instance of Owner with no pet */
    public Owner() { ... }
    public void addPet(Pet newPet) { ... }
}
```

(question continues on next page...)

(Question 11 continued...)

- i) **DRAW** a detailed UML class diagram containing ALL of the above classes and the interface, showing ALL relationships, packages, attributes and operations. (7 marks)

- ii) **NAME** the type of refactoring that is the most suitable to be applied to classes `Dog` and `CyberPet` in the given Java code. (1 mark)

- iii) **DRAW** the UML class diagram that results from applying the type of refactoring identified in part (ii).

You should suppress the details of the classes that are irrelevant to the refactoring. (4 marks)

- b) i) Making use of a diagram, **DESCRIBE** the sequence of activities involved in **Test-Driven Development**. (5 marks)

- ii) State **THREE** benefits of **Test-Driven Development**. (3 marks)

- 12.a) What are the TWO general approaches to dividing a software system into subsystems? Please GIVE a short explanation for each of them. (4 marks)
- b) DRAW a diagram to illustrate the **Model-View-Controller** (MVC) architecture. (6 marks)
- c) NAME the design pattern that is typically applied in the MVC architecture and BRIEFLY EXPLAIN the design pattern. (4 marks)
- d) NAME TWO Java classes from the `java.util` package that implement the design pattern in part (c). (2 marks)
- e) **Coupling** and **cohesion** are two key criteria used for measuring quality when designing object oriented (OO) software. BRIEFLY EXPLAIN ONE type of coupling and ONE type cohesion applied within OO software design. (4 marks)

END OF EXAMINATION PAPER