NATIONAL UNIVERSITY OF SINGAPORE

CS2103/T – SOFTWARE ENGINEERING

(Semester 1: AY2017/18)

Part 2

Time Allowed : 1 Hour

INSTRUCTIONS TO STUDENTS

☐ Please write your Student Number only. Do not write your name.

☐ This assessment paper contains **FOUR** questions and comprises **FIVE** printed pages.

☐ You are required to answer **ALL** questions.

☐ This is an **OPEN BOOK** assessment.

☐ You may **use pencils** to write answers.

STUDENT NO: _____

This portion is for examiner's use only

Question	Marks	Remarks
Q1	/4	
Q2	/4	
Q3	/4	
Q4	/4	
Total	/16	

Q1 [4 marks] Illustrate the class structure of the following code using a suitable UML diagram. In addition to the classes in the code, include the following classes:

- ProjectManager: Manages Project objects and Task objects
- StaffManager: Manages Person objects and Allocation objects.
- Data: Plays the role of a Façade class to sit in front of the others classes.

```java
class Project {
  List<Task> projectTasks;
  List<Person> projectMembers;
  //...
}

abstract class Task {
  /**
   * Tasks this task depends on.
   * A Task can depend on 0
   * or more Tasks.
   */
  List<Task> dependencies;

  abstract void start();
  //...
}

interface Chargeable {
  void charge();
}

class Person {
  //...
}
```

```java
class AdminTask extends Task
    implements Chargeable {
  @Override
  void start() {
    //...
  }
  @Override
  public void charge() {
    //...
  }
}

/**
 * An association class to represent
 * allocation of a Task to a Person.
 */
class Allocation {
  Person owner;
  Task task;
  Date from;

  //...
  Allocation(Person p, Task t) {
    //...
  }
}
```

Q2 [4 marks] Illustrate the interactions caused by calling the Modal#load() given below using a suitable UML diagram.

```
class Model{
   Storage s;
   private boolean isInit;

   void load(){
     Disk d = s.getDisk();
     if(isInit){
        d.getSize(this);
     }
   }

   public int getPeriod() {
     //...
   }
}
```

```
class Storage{

   public Disk getDisk() {
     return new Disk();
   }
}

class Disk{
   int getSize(Model m){
     return m.getPeriod() + 5;
   }
}
```

Q3 [2+2 = 4 marks] Assume you are testing the Foo#getClip() method given below.

```
class Foo{
  private String text;

  void setText(String text){
    this.text = text;
  }

  /**
   * Returns the first n characters of the text followed by "..."
   * if the text is shorter than n. Otherwise returns the original text.
   * e.g. If the text is "Apples", getClip(5) returns "Apple..."
   * and getClip(10) returns "Apples"
   * @param n must be a value in the range 10..100
   */
  String getClip(int n){
    //...
  }
}
```

a) Give *equivalence partitions* and *boundary values* for text and n. One example is given.

	Equivalence partitions	Boundary values
text	Longer than n	Size = n+1
n		

b) One test case is given. Suggest 4 more good test cases that are efficient and effective.

	text	n
1	Size = 10	10
2		
3		
4		
5		

4

Q4 [4 marks] Suggest at least 5 code quality improvements to the code below. You may ignore indentation issues.

Example suggestion

```java
private static int STATUS_STABLE = 0;
private static int STATUS_UNSTABLE = 1;
private static int UNKNOWN_STATUS = 2;

// ...

/**
 * Remove spaces and numbers, sort characters, and replace first
 * 10 letters with corresponding numbers.
 */
public string encode(String original) {
    logger.info("Entering the encode method");
    if (!original.isEmpty()) {
        original = removeSpaces(original);
        original = removeNumbers(original);
        original = sorter(original);

        //replace first 10 letters with numbers
        for (int i = 0; i < 10; i++) {
            original = original.substring(0, i) + getNumber(original.charAt(i))
                + original.substring(i + 1);
        }
    } else {
        return "";
    }
    return original;
}
```