UNIVERSITY OF LONDON

BSc EXAMINATION 2017

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2800: Software Engineering
CS2800R: Software Engineering – PAPER FOR RESIT CANDIDATES

Time Allowed: **TWO hours**

Answer ALL questions
Calculators are not permitted

1. For each of the following pairs of related software engineering concepts you must:

   - *Describe* carefully each of the two concepts.
     A good description could be about three lines of text. Enough to explain the concept to a new student on CS2800.
   - Show that you understand how the two concepts are *connected*.
     For example, they have the same or contrasting goals, or they may be techniques that rely on each other to work.

   (a) *Code Reliability* and *Metrics*. [10 marks]

   (b) *Black Box Testing* and *White Box Testing*. [10 marks]

   (c) *Code Smells* and *Refactoring*. [10 marks]

   (d) *Vertical Slice* and *Horizontal Slice*. [10 marks]

```
public static void CS2800_17(int val) {
  if (val %2 == 1) {
    gfl();
  } else {
    foo();
  }
  while (val %3 == 1) {
    val = foo();
    gfl();
  }
  bar();
}
```

CODE FOR USE IN ANSWERING QUESTION 2

2. (a) Describe three coding standards for the Java language that avoid faults. For each describe the type of fault that is avoided. [6 marks]

   (b) Sketch the flowgraph for the code shown above Question 2. [6 marks]

   (c) For the code code shown above Question 2, give example values for the argument `val` for a set of tests to cover all statements. [3 marks]

   (d) How is Cyclomatic complexity calculated for a piece of code and what evidence is there that restricting Cyclomatic complexity is a good idea?

   Illustrate your answer with reference to the code shown above Question 2. [3 marks]

   (e) Give two practical reasons why direct measures are preferred over indirect measures. [4 marks]

   (f) Give two examples of code attributes that cannot be directly measured and explain how indirect measures might be used. [4 marks]

3. (a) In the context of a Source Code Control System, what is locking? What problem does the lock-modify-unlock problem solve? [4 marks]

   (b) Using locking in a source code control system can cause problems in the development process. Describe two such problems. [4 marks]

   (c) Explain carefully the difference between a Working Copy of a project from an SVN repository and a local folder containing a copy of the files that make up a project. [2 marks]

   (d) What is a branch in the SVN system? Explain how the use of feature branches can support multiple software engineers working on different tasks at the same time within a complex project. You should carefully explain any processes that are required. [6 marks]

NEXT PAGE

4. Read the following description of a game program that needs to be updated.

   Each player controls an army made up of several units, each of which is at a location on a map, or in reserve.
   Each unit has a fighting power, an amount of gold and a health indication, all of which are positive integers. They also have a maximum health level.
   Units have a profession. The professions are Soldier, Medic and Animal.
   Units can be promoted. Each promotion increases the fighting power and maximum health. The promotion levels are positive integers.
   Units with zero health are dead and cannot fight, spend money or be deployed. Units can also be undead. Undead units cannot lose or gain health.

   The class `Army` represents an army, and the class `Unit` represents an individual fighting unit. Each instance of `Army` contains a collection of units.
   Each profession is represented by an extension of the `Unit` class.

   (a) Draw a UML class diagram of the classes described above. You should carefully draw (labeled) associations but *you should not include any attributes or responsibilities*. [6 marks]

   (b) For each of the following observations describe an *appropriate* design pattern that should be applied to the original design.

   You must give some details of how to apply the design pattern, possibly with a sketched UML diagram.

   i. The Unit class calculates damage taken during an encounter through a complex sequence of `if` and `switch` statements. [4 marks]

   ii. The different unit professions are essentially the same and we would like to be able to add more professions without cluttering the whole code base with references to these professions. [4 marks]

   iii. We need to extend the program to print a list of the number of units, at each promotion level, of each profession in the army. [4 marks]

END