UNIVERSITY OF LONDON

BSc EXAMINATION 2018

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2800: Software Engineering
CS2800R: Software Design – 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) *Vertical Slicing* and *Top-Down Integration*. [10 marks]

   (b) *Software Fault* and *Programming Style Rule*. [10 marks]

   (c) *Code Release* and *Incremental Business Value*. [10 marks]

   (d) *Too Few Comments Smell* and *Once and Only Once Smell*. [10 marks]

2. (a) What (in the course CS2800) are the two key properties of *good* code? [2 marks]

   (b) Software Engineering is made simpler by the use of appropriate tools. Briefly explain how THREE of the software engineering tools built into the Eclipse IDE support the writing of *good* code. [9 marks]

   (c) Describe THREE Java programming standards. For each state how it helps to ensure that you write *good* code. [9 marks]

   (d) We propose replacing SVN with a system called FallPackage.

   - Whenever we alter a file in our FallPackage folder it is automatically updated on the FallPackage server and in everyone else's FallPackage folder.
   - FallPackage tracks the username, date and time of each change.
   - FallPackage allows anyone to wind back any file to an older version.

   Give THREE reasons why a Software Engineer would find it problematic to replace SVN with FallPackage as their Source Code Control System. [6 marks]

3. (a) In a Source Code Control System (like SVN) what is a delta of a file. Why do we prefer negative deltas? [4 marks]

   (b) You are working on a new feature in a feature branch. You created your branch at version 34. Your branch is now at version 53. The trunk is at version 48.

   You have no working copy.

   You have not yet done a sync merge.

   i. What is a sync merge? [2 marks]

   ii. How does SVN decide which files to compare when sync merging? [2 marks]

   iii. Why should you perform a sync merge now? [2 marks]

   iv. What steps are required to perform the sync merge in SVN? [2 marks]

   v. What could cause a conflict during the sync merge, and what steps are required to resolve any conflict? [2 marks]

   (c) How can SVN be used to judge the the *productivity* of a particular software engineer? [4 marks]

   (d) How can SVN be used to judge the *quality of the code submitted* by a particular software engineer? [4 marks]

4. Read the following overview description of our product.

> At Counting Inc. we produce and sell a state of the art Linux based accountancy package.
> We incorporate an extensive indexing, sorting, filing and data persistence system (IndPers) from a third party company that we use to ensure speed and efficiency and to make sure that data is not lost even when the system goes down due to a power failure.
> We use the company's standard (complicated) API for networking (NETAPI) that is maintained by the central systems team.
> We have a graphics API (DrawIT) at the center of our accountancy package. We wrote this and its speed and beauty are key to our market success.

(a) We have a source code license for IndPers. Our lead developer Jim has decided to improve it so that it fits better into our code base.

i. Give TWO reasons why this is a bad idea. [2 marks]

ii. What structural design pattern is the ideal choice to solve this design issue: using third party software that we do not wish to change.
Explain the motivation for this choice of pattern. [2 marks]

(b) Different parts of our code perform the simple network operations we require. These classes include clever code written by Freda, our networks guru, for NETAPI. We do not all properly understand Freda's code.

i. Give TWO reasons why our use of Freda's code is a bad idea. [2 marks]

ii. What structural design pattern is the ideal choice to solve this design issue: simple use of a complex API.
Explain the motivation for this choice of pattern. [2 marks]

(c) We need to port our system, and particularly DrawIT, onto Android, Windows, IOS and iPhone. Each of these systems uses a very different set of graphics primitives. They all have windows, buttons, listeners etc., but they do it all very differently. To solve this we have massively simplified DrawIT so that it just paints everything on a Canvas, which works the same on all of the systems.

i. Give TWO reasons why this approach is a bad idea? [2 marks]

ii. What pattern supports coding windows systems on different platforms?
Explain the motivation for this pattern. [2 marks]

END