UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR08019 INFORMATICS 2C - INTRODUCTION TO SOFTWARE ENGINEERING

Saturday 15 th December 2018

09:30 to 10:30

INSTRUCTIONS TO CANDIDATES

Answer QUESTION 1 and ONE other question.

Question 1 is COMPULSORY. If both QUESTION 2 and QUESTION 3 are answered, only QUESTION 2 will be marked.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Convener: D.K.Arvind
External Examiner: J.Gibbons

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. THIS QUESTION IS COMPULSORY

   (a) Consider the OO design of a simple booking system for a hotel. Assume it provides a class for each of the italicised nouns in the following description.

   > The *hotel* has a number of rooms. Each *room* can be booked. A *booking* records the room booked, the *guest* who the room is booked for, the guest's arrival *date* and the guest's departure date. Guests register with the hotel, so their *personal details* do not have to be re-entered for each of their bookings. Functionality supported by the system includes asking if the hotel has free rooms for particular arrival and departure dates, booking a particular room, and generating a bill for a particular booking.

   Draw a UML class diagram showing the relationships between these classes, as appropriate showing a relationship between two classes using a typed attribute or an association. Include suitable multiplicities and ordering annotations on each association, but do not show navigability. Include suggested operations, but there is no need to show operation argument or return types. [*9 marks*]

   (b) i. When a program is *refactored* in what way does it change and in what way does it not change? [*2 marks*]

   ii. Why is refactoring particularly important with Agile software development processes? [*2 marks*]

   (c) Java classes implementing the `List<E>` interface for some element type `E` must implement a method

   ```
   E get(int index)
   ```

   for retrieving the list element at position `index`. Describe in English a suitable precondition and postcondition for this method. Remember list indexes in Java always start at 0 and the list method `size()` returns the number of elements in the list. Refer to the object reference that the method is called on as `this`. Such an object is both an input *and* an output of the method. [*5 marks*]

QUESTION CONTINUES ON NEXT PAGE

QUESTION CONTINUED FROM PREVIOUS PAGE

(d) You run a software consultancy with a team of 4 experienced software engineers. You are working on a bid for work with a local clock museum. The museum wants to have touch screen displays by its exhibits that enable visitors to see inside the clocks and understand how they work, as well as learn about their history. You have established that suitable hardware can be purchased off the shelf, but no existing museum information display systems on the market provide suitable software functionality. The museum would like to work closely with you on the software development and could organise museum visitors for evaluation.

   i. You realise that it would be appropriate to use a software development process that involves iteration and that can deliver a sequence of prototypes with functionality increasing and evolving from one iteration to the next. Explain why. [*3 marks*]

   ii. You know that both the Unified Process and XP (Extreme Programming) include an emphasis on iteration. Which of these two is preferrable and why? [*4 marks*]

2. ANSWER EITHER THIS QUESTION OR QUESTION 3

(a) You are designing the user interface for a software system to be used by the captain and senior crew on board large container transport ships to control their cargo operations. Two of the principles of UI design that you will follow are "User familiarity" and "Consistency". Name three of the others and give a short description of each.

You learn that some key operations on these ships include loading and unloading containers. How does the User familiarity design principle guide you in using this knowledge for your design? [*8 marks*]

(b) Consider a Java class for immutable objects representing distances in various units, including feet and metres.

```
public class Distance {

  public static Distance makeDistanceInFeet(double f) {...}
  public static Distance makeDistanceInMetres(double m) {...}
  public Distance getDistanceInFeet() {...}
  public Distance getDistanceInMetres() {...}
  public Distance add(Distance d) {...}
  ...
}
```

i. Write a JUnit test method that checks that the `add()` works correctly adding two distances in different units. Your test may assume that the test class statically imports the JUnit 4 method:

```
public static void
  assertEquals(double expected, double actual, double delta)
```

and defines values for constants

```
public static final double ONE_FOOT_IN_METRES;
public static final double DELTA;
```

where `DELTA` can be used for the `delta` argument of `assertEquals()`. Use good coding practices to write your test method in such a way that the units of all double values and the calculation of the expected result are clear. [*7 marks*]

ii. What does this three argument version of `assertEquals()` do and why is it preferred over the two argument version

```
public static void
  assertEquals(double expected, double actual)
```

? [*2 marks*]

QUESTION CONTINUES ON NEXT PAGE

QUESTION CONTINUED FROM PREVIOUS PAGE

(c) Previously you have gained experience with capturing functional requirements using use cases and now you are joining a team in a car company developing software for autonomous cars. Other team members have previously been most used to functional requirements described using lists of features and are curious about whether to try use cases instead.

i. Briefly explain the main advantage of use cases over traditional feature lists from the point of view of staff around the company who are consulted about requirements. [*2 marks*]

ii. How about from a testing point of view? [*1 mark*]

iii. Discuss whether you think use cases are appropriate, raising at least two potential problems with using use cases. Your answer should not just reproduce common issues noted in class. Some of these issues might be relevant to autonomous car software, in which case you need also to explain *why* they are relevant. Answers which raise other issues relevant to autonomous car software are also fine. [*5 marks*]

3. ANSWER EITHER THIS QUESTION OR QUESTION 2

(a) Consider the following description of a self-service bike-hire scheme for a city.

> When not in use, bikes are locked to *bike docks* located around the city. All bikes are connected via a mobile phone network to a central computer which can command bikes to lock and unlock themselves and can ask bikes to report their location.
>
> Before hiring a bike, users first must download an app to their smart-phone and buy a pass which allows them to hire a bike for up to 1 hour. When they are within 20m of some docking location with a free bike, they can signal via the app for the bike to flash its lights and to unlock itself. The user is then free to ride the bike for up to an hour.
>
> Bikes must be returned to bike dock locations and the app can be used to find locations with empty docks. When the bike is docked, the user signals via the app that the hire is finished and the system locks the bike. If the user wants a bike for more than 1 hour, one or more extension passes, each for 30 mins, can be purchased via the app.

Write out a structured use case for hiring and returning a bike, starting with signalling for a bike to be unlocked. Provide entries for the the fields *Primary actor, Supporting actors* (identify at least one), *Precondition, Trigger, Guarantee, Main Success Scenario* (MSS) and the *Extension scenario* involving hiring for over one hour. To make your scenario descriptions clearer, you may include steps which are not strictly to do with system interactions but are part of the wider business use case. You may find it useful to add remarks to your MSS description about how some steps or ranges of steps might be optional or repeated.

Consider users' phones, the mobile network, the central computer and the mobile transceivers attached to the bikes to all be within the system boundary, but the bikes themselves to be outside the boundary. [*9 marks*]

(b) When designing a class or software component it is desirable that it has *high cohesion.* What is cohesion and what are its benefits? [*4 marks*]

(c) i. Briefly describe one advantage and one disadvantage of a distributed version control system such as Git compared to a single repository system such as Subversion. [*3 marks*]

ii. What is a *workflow* for a distributed version control system? Why are workflows important? [*3 marks*]

QUESTION CONTINUES ON NEXT PAGE

QUESTION CONTINUED FROM PREVIOUS PAGE

(d) You work for a software company and are developing the UI for some new medical equipment. Your boss suggests you use a new graphics library that she has read about. You have heard of it too and know that it is supposed to have some very useful capabilities, ones that you had planned to implement yourself. You know the project is behind schedule and your boss is keen to see the UI finished; the customer is also keen for delivery. However, you are not sure whether the library has been well tested for reliability. You are in an ethical dilemma over whether to use it. Obviously you have an ethical duty to the client and your company to finish the UI as soon as possible. What are the three other ethical principles that come into play in this scenario? Name each one and describe how it applies in this context. [*6 marks*]