UNIVERSITY OF LONDON

BSc EXAMINATION 2023

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2800:Software Engineering

Time Allowed: **TWO hours**

Please answer **ALL** questions

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

   - Briefly *describe* each of the two concepts and why they are important in software engineering. This should be enough to introduce the concept to a new student on CS2800.

     The two descriptions could have points in common, which might then also be a part of their connection.

     Each description could be about six lines of text.

   - 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. This needs careful thought.

     A good answer could be about four lines of text.

   (a) *Literate Programming* and *Programming Standards*. [12 marks]

   (b) *UML class diagram* and *Design Patterns*. [8 marks]

   (c) *Too Many Comments Smell* and *Too Few Comments Smell*. [8 marks]

   (d) *Reintegration Merge* and *Sync Merge*. [12 marks]

2. (a) Why should programming standards be automatically checkable? [3 marks]

   (b) Give two common Java programming standards for variable naming. Both standards must be automatically checkable. [2 marks]

   (c) Consider the following code sample:

```
public float money;
private float interest;
/**
* @param value is how much money we want
* @return the number of years we need to wait
*/
public int Foo(float value) {
 int years = -1;
 if (value > 0) {
  for (years = 0; money < value; years = years + 1)
   value = value / interest ;
 }
 return years;
}
/*
 public void oldFoo() {
  int x = 3 ;
  while (x < 10) {
   x = x + 1 ;
  }
 }
*/

```

      i. Considering the Google Java programming standard as used in CS2800, identify three different standards failures in the code sample. Explain why each standard violation is important. [6 marks]

      ii. Identify two different code smells in the code sample. Explain how you would remove each smell. [6 marks]

      iii. Draw the flow graph for the method `Foo` from the code sample. What is the Cyclomatic Complexity of this method? [6 marks]

NEXT PAGE

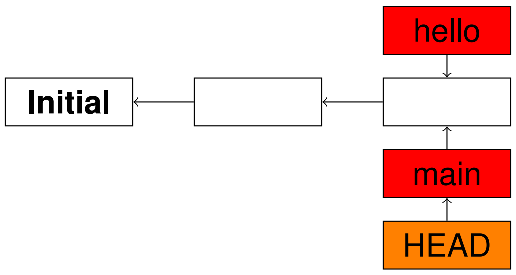
3. (a) How do branches keep track of different lines of development in the Git revision control system? [4 marks]

   (b) Give three advantages of using feature branches in a revision control system over committing everything to the main branch. Advantages seen by developers or by project managers or by both will get equal credit. [3 marks]

   (c) Give two reasons why it is not a good idea to immediately follow each `git commit` by a `git push`. [2 marks]

   (d) Your git repository contains no folders and just one file: 'Readme.md'.

   The following picture shows all of the branches (`hello` and `main`) and the current commits (there are three commits).



   The following lists of commands are done in order - first all of the commands from 3(d)i, then those from 3(d)ii, then 3(d)iii and finally 3(d)iv.

   List the files in the all branches, and draw a picture of the commits and branches, after each list of commands.

   i. `git checkout hello`
      create text file `bert` containing 'I am Bert'
      `git add bert ; git commit` [4 marks]

   ii. `git branch mybranch` [2 marks]

   iii. `git checkout main`
      create text file `bert` containing 'I am not Fred'
      `git add bert ; git commit ; git merge hello` [4 marks]

   iv. edit the file `bert`
      `git add bert ; git merge --continue` [2 marks]

NEXT PAGE

4. (a) The Singleton design pattern is a creational pattern that ensures that only one instance of a class can ever exist in the running code. Show how we can make a class into a Singleton in Java. [4 marks]

   (b) Explain why it is not desirable to extend (inherit from) a Singleton class, and describe the Java mechanism to prevent inheritance. [4 marks]

   (c) A Factory pattern can be used to build and return an instance, which can prevent two instances of a class being created. Explain how this might be done in Java and give two ways in which this approach is better than using a Singleton. [4 marks]

   (d) Even though the design patterns Facade and Adapter have the same UML class diagram, it is usually possible to determine which pattern was implemented by looking at the source code.
   Explain carefully how we can tell which of these two patterns has been implemented by examining the source code of an application. [4 marks]

END