Further Java - Supervisions

Alastair Beresford and Andrew Rice

There are two recommended supervisions for this course. Questions are marked up as:

- Bookwork (B): require review of lectured material and recollection of facts
- Shallow (S): direct application of facts in a formulaic manner
- Deeper (D): apply material in a new context or bring several ideas together
- Open (O): open-ended questions with no single (or no known) right answer

Those questions marked with an asterisk (*) are optional extension exercises.

Supervision 1 relies on knowledge of the syllabus taught by completing Ticklets 1 and 2. Supervision 2 builds on knowledge of the whole course.

Supervision 1

1.1 (B) In this course we adopted the Google Java Coding Standard. Describe why a coding standard is important.

1.2 (S) List at least 10 examples of poor style in the following Java code and describe *why* they are problematic and *how* they could be fixed:

```
import java.util.*;

public class wrong {
  public static final boolean t = true;
  public static final boolean f = false;

  public static void oldness(Integer o)
  {
    boolean b = o > 0;
    if (b == f)
    System.out.println("Input must be positive");
    else
        System.out.print("Age in days = ")
         System.out.print(o * 365.25);
  }
```

```
  static public void main(String args[]) {
    Integer i = Integer.parseInt(args[0]);
    oldness(i);
  }
}
```

1.3 (S) Write two simple Java programs which exhibit the two common patterns for creating and executing a new Thread in Java. Compare and contrast the advantages and disadvantages of each approach.

1.4 (D) Write a client for the *finger* protocol (RFC 742) which is able to remotely retrieve information about users of a Unix system. Your solution should take a user and domain as a single argument, for example "`java Finger arb33@hermes.cam.ac.uk`" and print out details of the user `arb33` on `hermes.cam.ac.uk`.

Note: Your computer will need to be on the University network to successfully connect to `hermes.cam.ac.uk`. You can test whether your computer is able to connect by using the `finger` command-line app found on Mac OS and many Linux systems.

1.5 (S) Write down a list of all the ways in which your implementation for Question 1.2 above might throw a Java Exception. Describe, in words, the necessary steps required to ensure your solution is robust to failure.

1.6 (S) Swap your solution for Ticklet 1 with your supervision partner. Highlight all the ways in which your partner's solution for Ticklet 1 does not adhere to the BAE coding standard. In what ways could the legibility, robustness and correctness be improved?

1.6 Complete sections (a) and (b) from 2010 Paper 5 Question 9.

1.7* (O) Write a simple Android app which acts as a Chat Client as described in Workbook 1 or 2.

1.8* (O) Research and describe in words what a *serialization bomb* is. Explore whether you can devise a means of preventing a serialisation bomb from quickly exhausting all available resources of the Java virtual machine.

Supervision 2

2.1 (B). Describe why the definition of Vector Clock algorithm, as found in this course, is different from that found in the Part IB Concurrent and Distributed Systems.

2.2 (B). What are the rules for updating a Vector Clock?

2.3 (S) Complete Table 1 in Workbook 3 and explain the result.

2.4 (S) A student attempts to implement fine-grained locking to provide atomicity for the `transferTo` method of the `BankAccount` class as described in Workbook 3. Their solution is as follows:

```
class BankAccount {
  private int balance;
  private int account;
  public void transferTo(BankAccount b, int amount) {
    synchronized(this) {
      balance -= amount;
    }
    synchronized(b) {
      b.balance += amount;
    }
  }
}
```

Describe why this solution is incorrect and why testing for correctness is hard. Describe in words how to write a correct implementation. Provide your supervisor with a copy of your work for Ticklet 3.

2.5 (D) Complete the following exam questions:

- 2010 Paper 3 Question 9
- 2011 Paper 3 Question 7
- 2013 Paper 3 Question 7
- 2016 Paper 3 Question 6

First, write you solutions on paper. Then type up your answers and test them on a computer. What errors did you make when writing solutions on paper, and how could you avoid making them in the exam hall?

2.6* (O) Java serialization allows malicious clients to subject a server to a denial-of-service attack (a *serialization bomb*; see Question 1.8). One alternative is to use a different encoding format (e.g. JSON, Protobuf) for messages sent between the client and the server. Rewrite your implementation of the Chat Client and Chat Server to use an alternative encoding format for messages. We strongly recommend you make use of an existing library to support encoding and decoding messages.

2.7* (O) The Java NIO libraries provide support for non-blocking sockets. This means you can test whether data is available on a socket without blocking. This will allow you to write a

variation of the Chat Server which only requires a single thread (rather than 2n+1 where n is the number of clients). Your task is to research and implement a new Chat Client and Chat Server which does this.

Hint: If you decide to use serialization to support sending and receiving of messages then you will need to encode and send the payload length separately before the object so you can determine when you have received enough data that you can successfully read an object without blocking. Alternatively, you may choose to use a different encoding format (e.g. JSON, Protobuf) for the messages sent between the client and the server.