University of Stirling
Computing Science and Mathematics
CSCU9A2 Tutorial 3 Spring 2017
(for week starting 13 February)

1. This question is about thinking about the *design* of an interactive application, with a *GUI* and *events* arising from the user's interaction with the application:

   Consider implementing a simple word processor, rather like Notepad:

   - The document being edited is a single (potentially very long) String of characters, all displayed in the same font and style, in a text area (in Java a JTextArea is like a JTextField with many rows).
   - The cursor is always located at one position in the text, *between two characters, or before the first character or after the last character*.
   - The Delete key deletes the character immediately after the cursor (if there is one), and the Backspace key deletes the character immediately preceding the cursor (if there is one).
   - Pressing any ordinary keys (letters, digits, symbols) inserts the character at the current cursor position and the cursor is then positioned after the new character.
   - We are able to click with the mouse to indicate a new position for the cursor.

   The screen display is "dumb" – it does not itself keep a track of what the text is, nor where the cursor moves, and does not itself carry out insertion and deletion – it is up to our program to react to user generated events and to update and redisplay the text and cursor symbol when necessary.

   Design a set of internal Java variables for holding the current state of the document and any other information that is necessary in order to be able manage the text and display. Think about which GUI events the program must react to, and outline what action should be taken as the appropriate reaction – focussing on any changes to the Java variables. ***No detailed coding is required.***

2. Consider a simple Java method that prompts the user to enter their age in years, and then displays various statements about the status of the user depending on their age using `System.out.println`. Here is the `checkAge` method, which involves several `ifs` (with one small logical fault introduced here):

```
public static void checkAge()
{
   int age = readInteger("Please enter your age:");
   System.out.println("Your age is " + age);
   if (age >= 16)
   { System.out.println("You may marry"); }
   if (age >= 18)
   { System.out.println("You may vote"); }
   if (age > 6)
   {
      if (age <= 16)
      { System.out.println("Junior rate bus travel"); }
      else
      { System.out.println("Adult rate bus travel"); }
   }
   else
   { System.out.println("Free bus travel"); }
}
```

PTO

Here is a specification in English of the task that this method was designed to perform:
"The status of a person depending on their age is reported: Children up to and including 6 years old travel free on the bus, but are not allowed to marry nor vote. Children from 7 – 15 inclusive pay junior rate on the bus, but are not allowed to marry nor vote. 16 and 17 year olds may marry, and pay adult rate on the bus but may not vote. People aged 18 or over may marry and vote, and pay adult rate on the bus."

(a) What are "black box" tests? Propose a set of "black box" tests for this method, and explain why they are appropriate.

(b) What are "white box" tests? Propose a set of "white box" tests for this method, and explain why they are appropriate.

(c) Which of your tests from (a) and (b) would expose the logical fault in the program?

SBJ February 2017