University of Stirling
Computing Science and Mathematics
CSCU9A2 Tutorial 2 Spring 2017
(for week starting 6 February)

1. Programming for the Brookshear machine:

   Here is a fragment of a Java program:

   ```
   int total = 0;
   int i = 1;
   while (i != 11)
   {
      total = total + i;
      i = i + 1;
   }
   ```

   (a) Describe the result that is being computed in the variable `total`.

   (b) Sketch a *flow diagram* for this program.

   (c) Design a Brookshear *assembly language* implementation of this Java fragment.

   (d) Convert your Brookshear assembly language implementation into hexadecimal machine language – assume that the code will be loaded starting at memory address 20 (in hexadecimal), and you should choose suitable memory locations for the variables.

2. Review of using arrays in Java:

   Given that we have an array called `list` created like this:

   ```
   int[] list = { 88, 12, 6, 99, 4, 5 };
   ```

   (a) Trace through the following fragment of Java *in detail* to see what effect it has on the array:

   ```
   for (int i = 0; i < list.length-1; i++)
   {
      if (list[i] > list[i+1])
      {
         int temp = list[i];
         list[i] = list[i+1];
         list[i+1] = temp;
      }
   }
   ```

   (b) Draw a *flow diagram* for the operation of this fragment of program.

   (c) Why would it be inappropriate for the loop index variable `i` to count up towards `list.length` rather than `list.length-1`? What would happen if the Java were altered to count up towards `list.length` instead?

   (d) What would be the overall effect of repeating execution of this fragment of Java several times (say, as the body of another loop)?

SBJ February 2017