PAL
Consolidation 2

University of Hertfordshire
– Week 4 – October 2015 –

1 Aim & Objectives

- Re-cap on previous session
- Questions & Answers on previous sheet
- Work through exam related questions

2 Introduction

In this session we will be concentrating on ideas and concepts related to the up and coming exam, where we plan to cover topics that have arisen from previous years. Please note that this session is not a replacement of doing your own revision and should only compliment the work you do in your own time!

3 Keywords

Condition – State or values of a system. It describes all the parts that are able to change or hold value within the algorithm or code.

Pre-condition – "*Pre*" means prior of before. This is the condition of the program prior to running it.

Post-condition – "*Post*" means after, so this means the condition after the algorithm or code has been run.

4 Exercise - Trace Tables

These are expected to appear in your test as they have in previous years. Trace tables teach an important way of thinking - the ability to see what a program is doing and thinking step-by-step. If you can create trace tables, you should be able to create these (eventually) in your head and therefore be able to work out what a program is doing without running it.

Generate a trace table for the following algorithm:

```
1 START
2   count = -1
3   WHILE count < 5
4     DISPLAY count
5     count += 2
6   ENDWHILE
7 STOP
```

Please use some paper and a pencil if you have these.

Step	count	<condition >	output
1			
2	-1		
3		true	
4			-1
5	1		
6			
3		true	
4			1
5	3		
6			
3		true	
4			3
5	5		
6			
3		false	
7			

5 Exercise - Trace Tables Advanced

We will now make these trace tables a little more difficult. If you are able to complete this exercise well then you should have no issues with trace tables, although you should still revise to keep your memory fresh closer to the exam.

When does this program stop?

```
START
  alpha = 3
  beta = 3
  WHILE (alpha >0) && (beta >0)
    DISPLAY beta
    beta -= 1
    IF beta <= 0 THEN
      alpha -= 1
      beta = alpha
    ENDIF
  ENDWHILE
STOP
```

Please make sure you write out your answer in a clear table so that somebody else (whoever is marking your workings) can read it!

In display, the output should be:

3, 2, 1, 2, 1, 1

'beta' should follow a similar pattern, with alpha decrementing each "*cycle*".

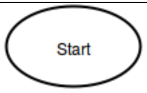
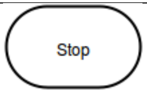
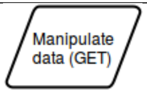
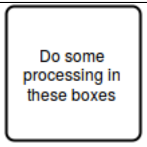
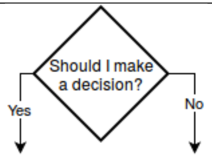
What would the program 'DISPLAY' if 'alpha = 5' and 'beta = 5'? (This is a test of whether you understand the code!)

5, 4, 3, 2, 1, 4, 3, 2, 1, 3, 2, 1, 2, 1, 1

6 Flow-Charts

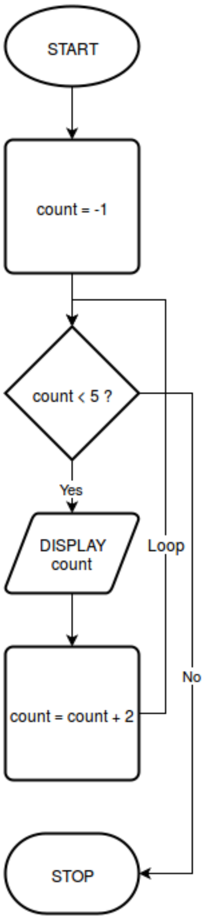
Flow-charts are very similar to pseudo code except they show a more visual way of imagining the program. Typically these are used for showing how decisions may happen in an algorithm or code, hence they may also be called "*decision charts*".

The basic blocks you will need are as follows:

Symbol	Meaning
	This represents an entry point into the chart. Typically there is only ever one of these, although subroutines (continuation or sections that will be re-used) may use multiple of these.
	There should only usually ever be one of these, although it's no unheard of for there to be more than one exit from an algorithm or function. It's usually bad practice, so try to avoid this if possible.
	These are used to manipulate data. You should only be using these to 'GET' in the simple exercises you'll be doing.
	These typically define something that needs to be done. You need to be sensible with this one, writing "*Do all the work*" is not a get out clause to drawing this diagram, you still don't understand what needs to be done.
	This is where your choices are made in the program. You'll find almost all diagrams have two exits from this diamond shape, 'true' or 'false', or in this case 'yes' or 'no'. These are what put the "*flow*" into flowcharts.

7 Exercise - Draw Flow-Chart

Now it's time to put your skills into practice, draw the flowchart for the pseudo code in the first exercise. Again, draw this on paper with a pencil.



8 Resources & Further Reading

'*http://homepages.herts.ac.uk/~db12aba/*' – All content from these sessions updated weekly.

'*http://code.org/*' – A good resource testing your programming skills.

'*http://stackoverflow.com/*' – Highly recommended online help for programmers (NOTE: Employers are interested to know whether you're an active member of this site!).

'*http://www.draw.io*' – A very good, free online drawing tool that exports to many formats, including 'XML' and 'JPG'.