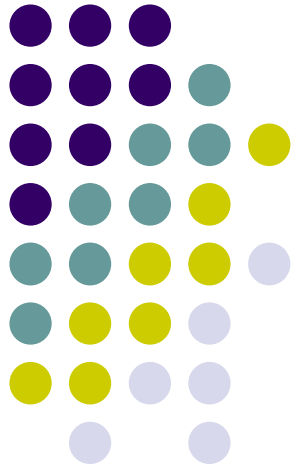
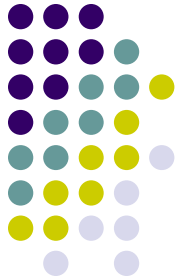
Structured Programming The Basics

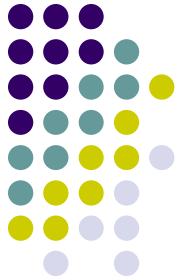


Control structures



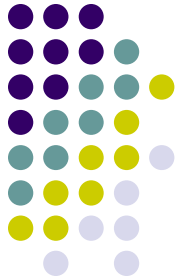
- They control the order of execution
- What order statements will be done in, or whether they will be done at all (skipping)
- Different from **data structures** – which are ways to access data, to operate on it

Why do structured programming?



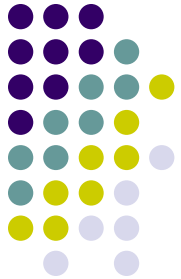
- It's easier to understand code written using structured programming
- Easier to test and debug code
- Easier to modify and maintain code
- Easier to work with other people to write large programs

4 Control Structures



- Sequence
- Selection
- Iteration
- Module

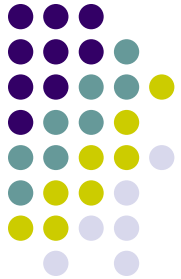
Guarantees for All Structures



- ONE Entrance
- ONE Exit

SEQUENCE

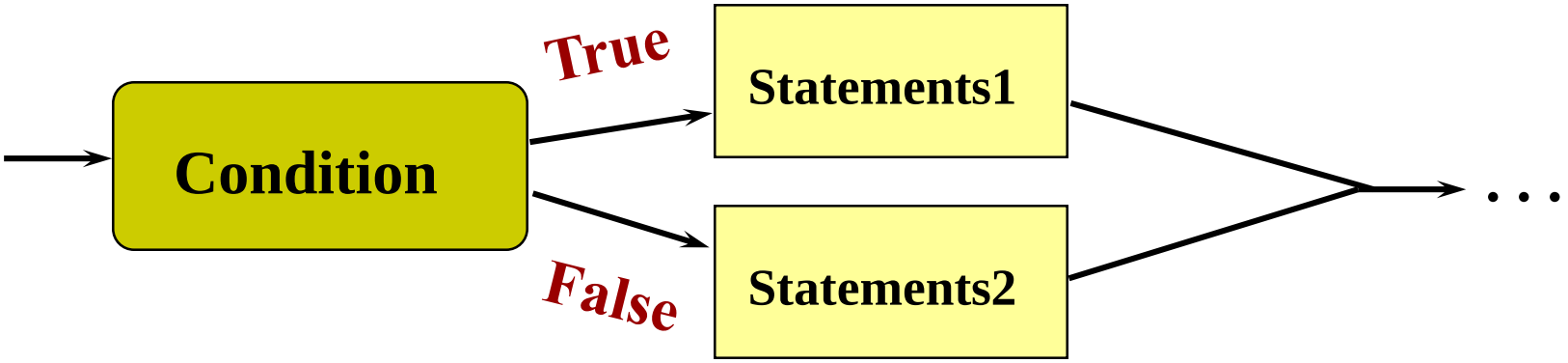
Guarantees for Sequences



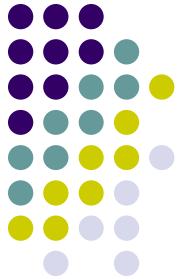
- Will execute the steps in the order given
- Will not enter or leave sequence in mid-stream
- Will not skip steps

SELECTION(branch)

IF Condition THEN Statement1 ELSE Statement2

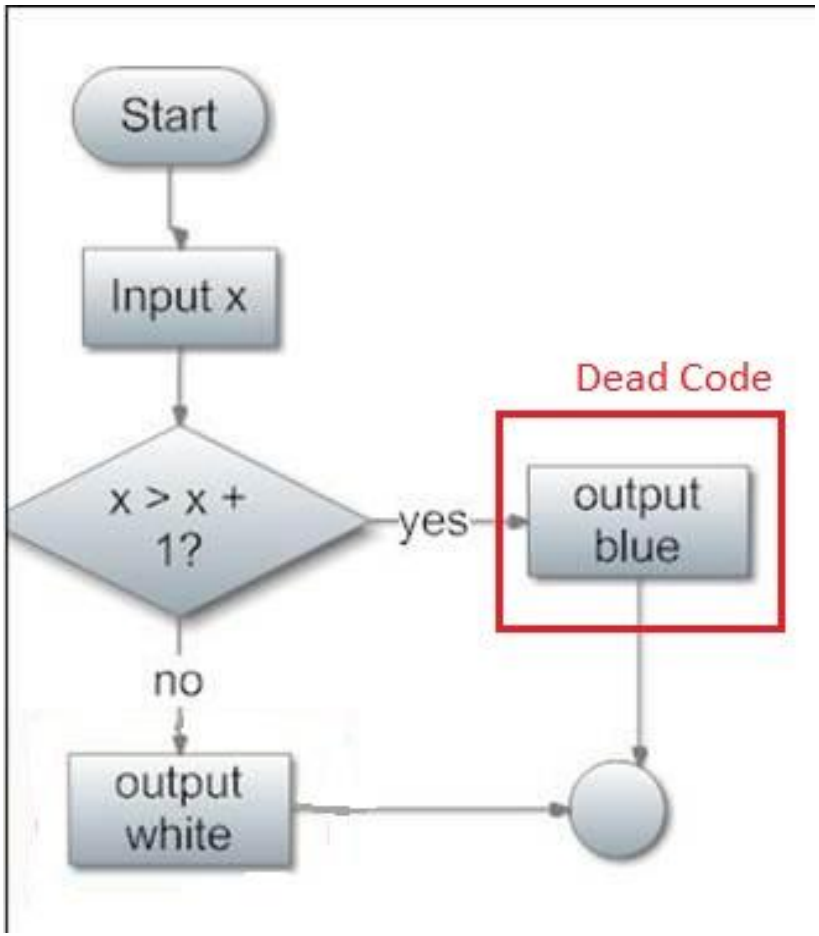


Selection Guarantees



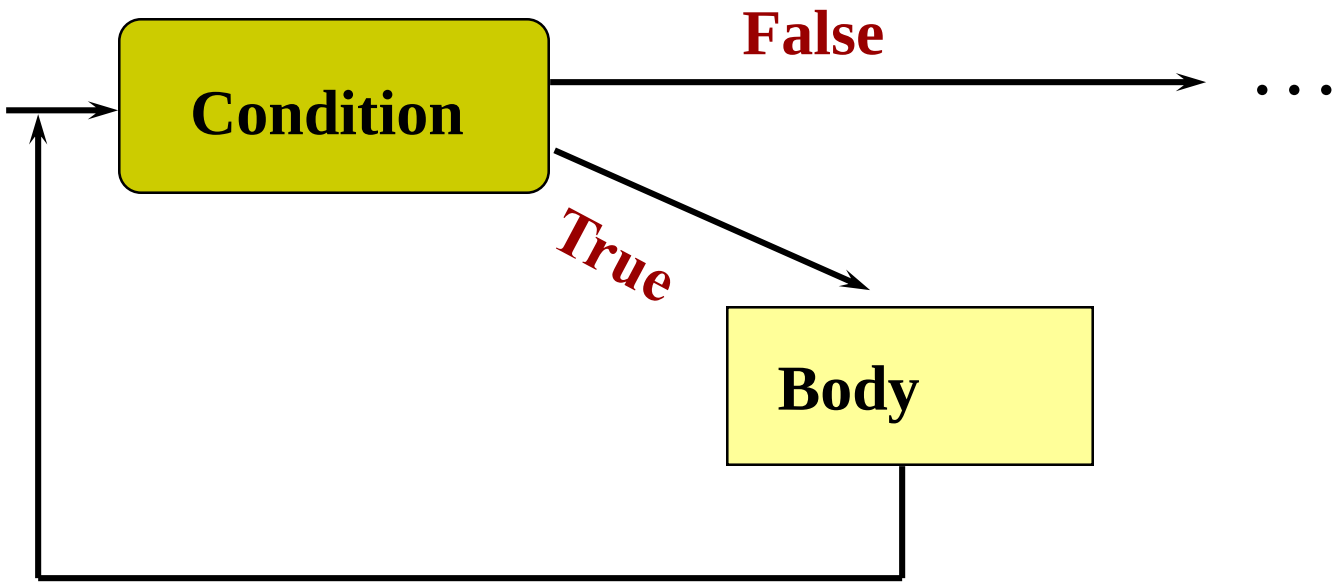
- Control always enters through the condition / question
- One branch or the other is executed, never both on one run
- MUST execute one branch or the other
- Processes in branches can be as large or small as you want
- Do not write Dead Code!

Dead Code

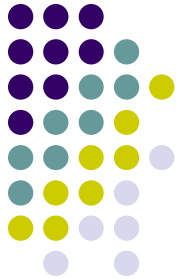


LOOP(repetition)

WHILE Condition DO Statement1

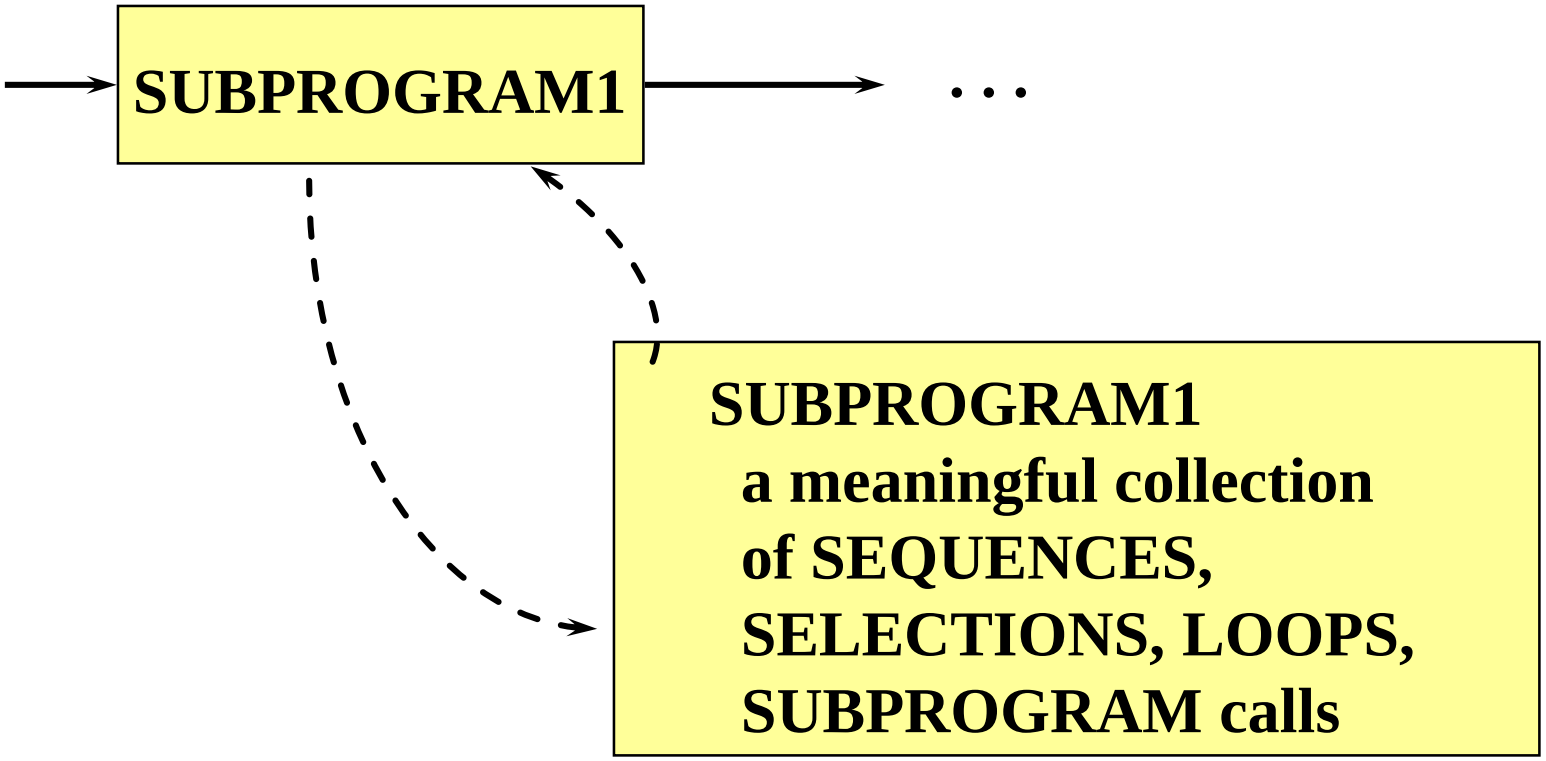


Guarantees



- Will go through test / condition at top to get into loop
- ALL of body will be executed before test is done again
- Body will be repeated until test is answered differently (NO)
- Do not write Infinite Loops!

SUBPROGRAM(function)



Module Flow of control