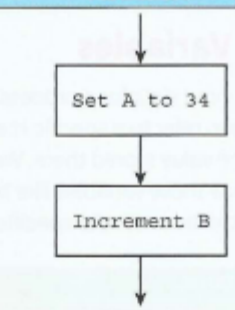
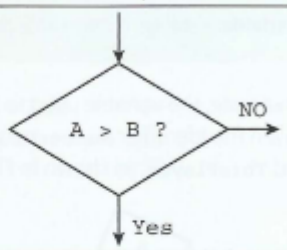
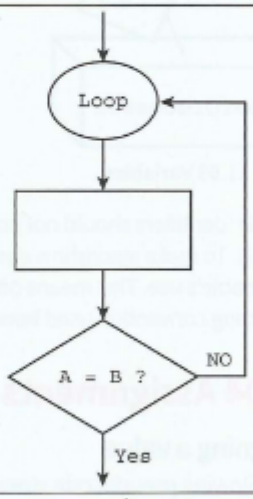
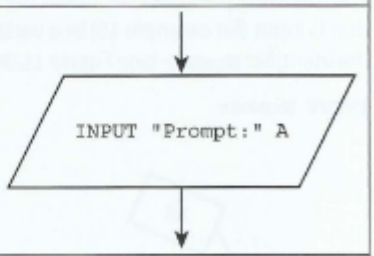
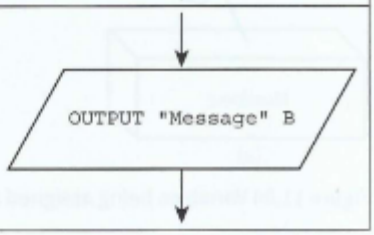
Paper 2: Theory

Algorithm and Problem solving

- Algorithm is a sequence of steps that can be carried out to perform a task.
- Algorithm is a step by step solution to a problem.
- A good algorithm should be:
    - Concise (i.e. to the point)
    - Not redundant (i.e. no repetition(this doesn't mean lack of loop))
    - Non ambiguous (i.e. step should be clear with no ambiguity)
- Algorithm can be expreessed in 3 ways:
    - Structured english: Steps in plain english
    - Pseudocode: by using keywords, identifiers without following syntax of a language but making coding easier in future.
    - Flowchart: Graphical representation of sequential steps.
- Programming constructs when writing algorithm:
    - Assignment: Associating value to a identifier
    - Sequence: Number of steps performed one after another
    - Selection: Execution of different steps based on some condition
    - Repetition(Loop/): Performing sequence of steps a number of time.

> Every solution involves inputting some data, processing it and outputting results.

	Structured English	Pseudocode	Flowchart
Assignment and Sequence	SET A TO 34 INCREMENT B	A ← 34 B ← B + 1	
Selection	IF A IS GREATER THAN B THEN ... ELSE ...	IF A > B THEN ... ELSE ... ENDIF	
Repetition	REPEAT UNTIL A IS EQUAL TO B ...	REPEAT ... UNTIL A = B	
Input	INPUT A	INPUT "Prompt:" A	
Output	OUTPUT "Message" OUTPUT B	OUTPUT "Message" B	

- Identifiers are names given to entities.
- Variables are entities whose value can be changed. It is storage location with an identifier.
- Swap:

```
Temp <- Value1
Value1 <- Value2
Value2 <- Temp
```

- Logical Statements (Boolean statement/ Conditions): Statements whose values can be true or false.

> Nested if/ Nested conditionals: One conditional statement within another.

> Nested loop: One loop within another

> Rogue value: is a value used to terminate a sequence of values.

Types of loop:

- count-controlled/repetition-controlled loop
    - FOR .... NEXT loop
- pre-condition/entry-controlled loop
    - WHILE ... [DO(optional)] .... ENDWHILE
- post-condition/exit-controlled loop
    - REPEAT ... UNTIL

> Each loop can perform what other loops can.. but choosing proper loop ensures efficient and managed code

Arrays Vs list:

- Arrays are sequential collection of homogeneous(same data types) values
- Lists can be heterogeneous (different data tyeps).

> Python has lists, not array

Searching:

- Process of finding data in a ccollection
- Two search algorithms:
    - Linear search: Searching one by one in array. Doesn't require sorted data.
    - Binary search: Searching by diving the search space into half with each iteration. Requires sorted data.

Sorting:

- Process of arranging a collection of values in a definite order.
- Commonly sorted in ascending or descending order.
- Two of the sorting algorithm:
    - Bubble sort:
        - Iteratively bubble out(i.e. take to end) largest, second largest.. element towards end.
    - Insertion sort:
        - Insert each element to appropriate position iteratively.
- Bubble sort

```
n <- MaxIndex - 1
FOR i <- 1 TO n
    FOR j <- 1 TO n
        IF MyList[j] > MyList[j+1] THEN
            Temp <- MyList[j]
            MyList[j] <- MyList[j+1]
            MyList[j+1] <- Temp
        ENDIF
    ENDFOR
ENDFOR
n <- n - 1
```

- Insertion sort in paper 3

Stepwise refinement

> Breaking down the steps of an outline solution into smaller and smaller steps

- Decomposition: Process of breaking down problem into sub-problems. Each sub task after decomposition is turned into module.
- Procedure: Group of task that performs number of steps. Steps in procedure can be executed by calling the procedure by its name.
- Function: A sequence of steps that is given an identifier and returns a single value. Function call is part of an expression.

Features of good pseudocode:

- Proper spacing
- Proper indentaiton
- Proper identifers (function name, variable name, constant name)
- Comments where required
- KEYWORDS capitalized

Modules

> Modules are block of code that perform an action. If the block of code produces and provides an answer(return value), it is called function. If it doesn't it is called procedure.

	Function	**Procedure**
Definition	A sequence of steps that is given an identifier and returns a single value.	Group of tasks that performs a number of steps.
Return Value	Returns a single value.	Does not return a value.
Call	Function call is part of an expression.	Procedure can be executed by calling it by its name.

	Function	Procedure
Use Case	When a task needs to produce a result that will be used in further computations.	When a task needs to perform an operation without needing to return a value.

- Argument: An argument is the actual value that is passed to a function when it is invoked/called. For example, in add(2, 3), 2 and 3 are arguments.
- Parameter: A parameter is a variable used to refer to that value inside the function's definition. For example, in the function definition function add(x, y) { return x + y; }, x and y are parameters.

Pass by value vs Pass by reference

> Pass by value and pass by reference are two methods of providing data to a function.

- Pass by Value: A method where a copy of the data is passed into the function.
- Pass by Reference: A method where a reference to the data is passed into the function.

Software development life cycle

> Software development life cycles are formal stages of software development

Stages in SDLC:

- Analysis(Understanding the problem):
    - Requirement specification
    - Feasibility study
- Design(General plan):
    - Decomposition
    - Choosing approach
    - Structure diagram, state-transition diagram
    - Algorithm
- Coding(Programming):
    - Program or set of programs is written
- Testing(Checking if it works):
    - Run to find and resolve issues
- Maintenance(After deployment):
    - Corrective maintenance: Fixing errors
    - Perfective maintenance: Improving performance
    - Adaptive maintenance: Adding features

Types of testing:

When the program is under development:

- White-box testing: Testing logic of each path through a program using the code.

> White-box testing involves dry-run/walkthrough.

- Black-box testing: Testing just the input and output without seeing code.
- Unit testing: Testing each module individually
- Integration testing: Testing if modules work together properly.
- Stub testing: If the modules are not implemented completely, integration testing can be done using dummy modules(stub).

After development:

- Alpha testing: Test done by the development team, mostly Testers/Quality-Assurance team.
- Beta testing: Program is released to small user base.
- Acceptance testing: Program is provided to customer who checks if the requirements are met.

Test data

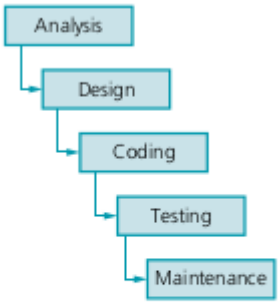
> Values used during testing

- Normal test data: Data that program should accept and work properly with
- Abnormal test data: Data that the program should reject
- Extreme test data: Edge cases or limit of data. Also called boundary test data. (Upper and lower limit)

Commonly used SDLC:

- Waterfall model:

> Waterfall model takes linear approach where each stage is completed before next is started



▲ **Figure 12.2** The waterfall model

- Advantages:
    - Easy to manage, understand and usee
    - Stages don't overlap

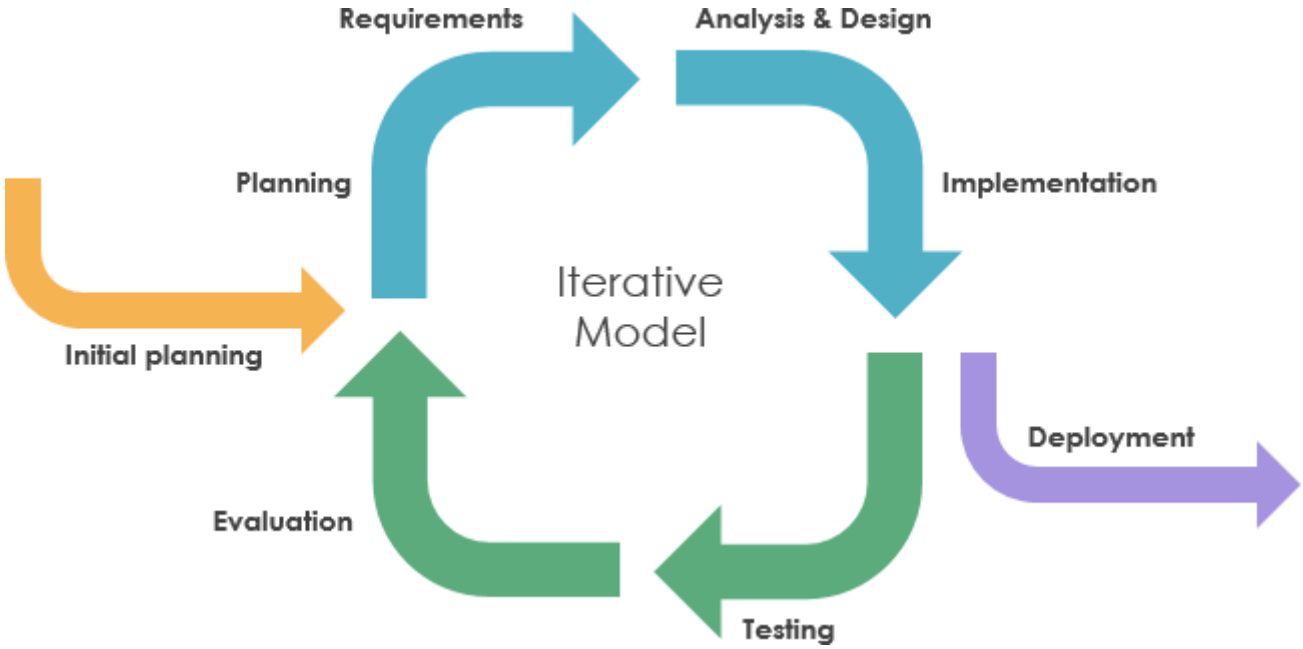
    - Specific goals and tasks in each stage
    - Good for small programs with well defined requirements

- Disadvantages:

    - Difficult to change requirements at later stage
    - Working program developed late in lifecycle
    - Not suitable for long, complex projects

- Iterative model

> In iterative model, simple subset of requirements is develeoped and features added with each iteration.



- Advantages:

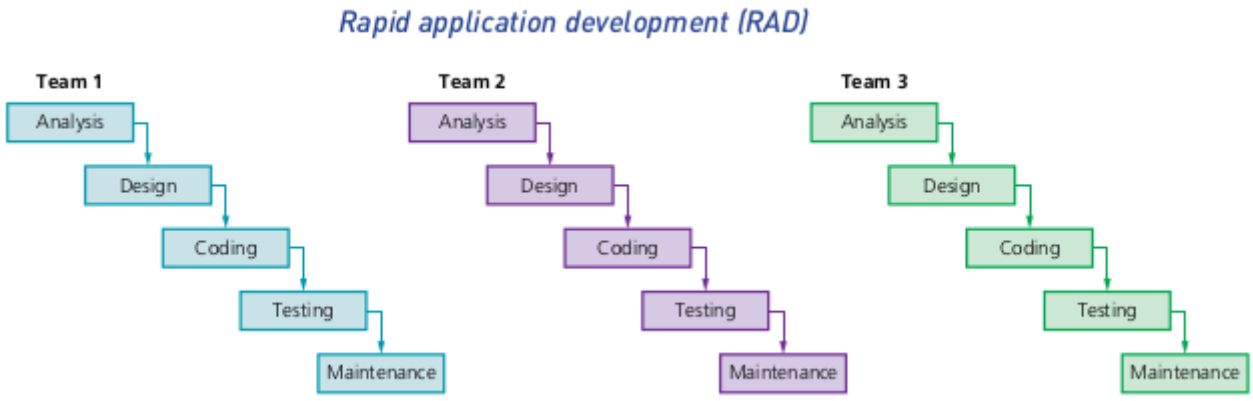
    - Working program at early stage
    - Flexible to requirement changes
    - Customer involved at each iteration so no surprise at end
    - Easier debugging and testing

- Disadvantages:

    - Whole system needs to be defined at start so that development can be broken down.
    - Needs good plannig and supervision
    - Not suitable for short and simple projects

- Rapid Application Development (RAD) model

> RAD focuses on code resuse and fast delivery with parallel development



- Advantages:
    - Reduced development time
    - Customer feedback is frequent
    - Modification, debugging and testing easier
- Disadvantages:
    - Software needs to be modular
    - Skilled developers required
    - Not suitable for short and simple projects

Data types and Structures

- Data types are classification of values used in programming.
- Primitive data types: Data types built in the programming language
- Composite data types: Data types created by combining multiple data types.

Records:

- Records are composite data types formed by the inclusion of several related items that may be of different data types.

> Records are composite user-defined heterogeneous data types

```
TYPE <NameOfRecord>
    DECLARE <identifier> : <data type>
    DECLARE <identifier> : <data type>
    ::
    ::
END TYPE
```

Arrays:

- Arrays are collection of items of same data type

- Index: Value used to speecify each item in an array.
- Lower and Upper bounds are index of first and last element in array respectively.
- 1D arrays have array in a single row.
- 2D arrays have rows and columns like table.

Abstract data types:

> Abstract data types are collections of data and the operations used on that data.

- Some ADTs are:
    - Stack: First In Last Out (FILO) based data type. Data is inserted and removed from same end
    - Queue: FIFO data type. Data is inserted and removed from opposite end.
    - Linked list: Collection of nodes made up of data and pointers.

Library:

> Libraries are collection of functions packaged together

- Advantages:
    - Well tested
    - Well documneted
    - Optimized (mostly)
    - Easy to use
- Disadvantages:
    - Not tailored/customized to our need
    - Not optimized for specific situation.