Week-1

Datasets [eg: scores, shopping bills, words etc.]

Iteration: Going through a sequence of objects and performing the same operations on each object. For ex: counting.

Variable : An entity whose value keeps changing as the computation goes on.

Filtering : To take out specific type of data from entire dataset.

→ The pattern of doing something repetitively is called an "ITERATOR"

Description of the iterator

Initialisation step: arrange all the cards in an "unseen" pile

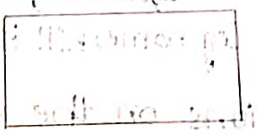
Continue or Exit: If there are no more cards in the unseen pile, we exit otherwise we continue.

Repeat Step: Pick an element from the "unseen" pile, do whatever we want to with this element, and then move it to another "seen" pile.

Go back to Step-2.

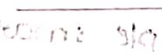
Introduction to Flowcharts

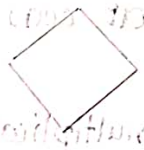
Some commonly used symbols



process or activity
set of operations that change the value of data (variables)

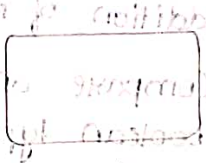
Flowline or Arrow
It shows the order of execution of program step





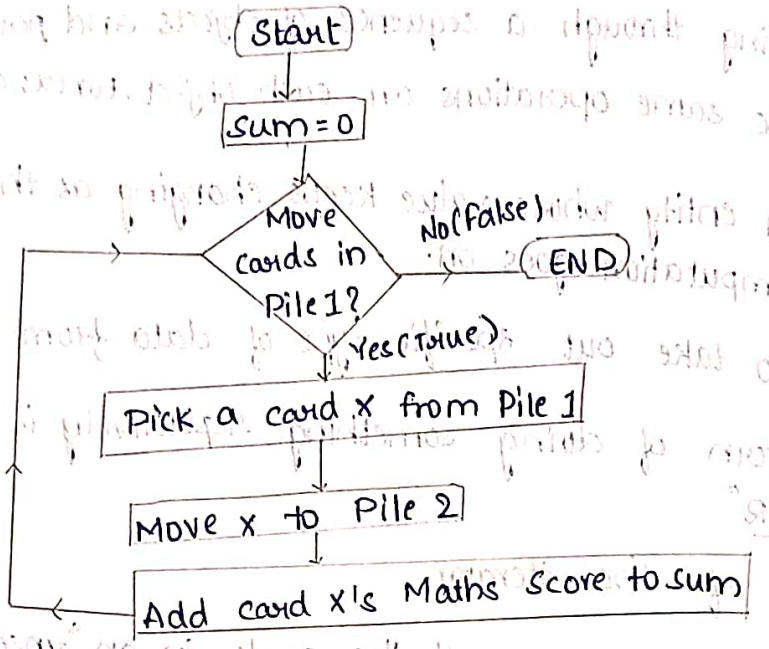
Decision
Determines which path the program will take
[condition]

Terminal
Indicates the 'start' or 'End' of the program



Some flowchart examples

↦ To find total maths marks of all students from mark sheet.



Modifications

- If we need only boys marks
- If we only need girls marks
- If we need boys & girls marks separately

Sanity of Data

- We organised our data set into cards, each storing one data item
- Each card had a number of elements e.g numbers (eg marks) sequence of characters [eg. name, bill items etc]
- We observed that there were restrictions on the values each element can take:
  - → for example: marks has to lie b/w 0 and 100
  - → name cannot have funny characters
- Constraints are the kind of operations that can be performed:
  - → Addition of marks is possible (but not multiplication)
  - → Compare one name with another to generate a boolean type (T or F)
    (but can't add a name with others)

This leads us to the concept of Data Type...

→ By associating a DATA TYPE (or simply type) with a data element, we can tell the computer (or another person) how we intend to use a data element.

- What are the values (or range of values) that the element can take?
- What are the operations that can be performed on the data element?

→ When we specify that a variable is of a specific type, we are describing the constraints placed on that variable in terms of the values it can store, and the operations that are permitted on it.

BASIC OPER DATA TYPES

- Boolean Has two values True or False
  operation: AND, OR, NOT
  result type: Boolean
- Integer Range of values is ..... -3, -2, -1, 0, 1, 2, 3 ....
  operation: +, -, ÷, ... <, >, =
  result type: Integer ..... ; Boolean
- Character values → alphanumeric
  ABC .... Z ab... z 0 1 2 9
  Special characters:
  . , ; : * / & % $ # @ !
  operation: =
  Result type: Boolean
- String values - any sequence of characters
  operation: char in string? ; =
  result type: Boolean ; Boolean.

Subtypes of Integer

SeqNO
- Range of values is: 0, 1, 2 .... Max
- Max can be some reasonable number, eg 10000 which is the largest data set size that we can handle.
- None of the integer operations make sense for the SeqNO data type

Marks Range of values is 0, 1, 2, ... 100

operation	result type
+, -	marks
<, >, =	Boolean

Count Range of values is 0, 1, 2, 3 ....

operation	result type
+, -	count
<, >, =	Boolean

x, ÷ do not make sense for counts

Subtype of character data type:

Gender values: M or F

operation	Result type
=	Boolean

Subtypes of String data type.

Names Values are strings with no special characters /city

Words Values are strings with alphanumeric and ' , ; : .....

Category Can take only one of the following values: "Noun", "Verb", "Preposition", "Adjective"....

Transformation of Data types

Date: Subtype of integer

Range of values is 0, 1, 2, 3 ... 365

operation	Result type
print	string
<, >, =	Boolean

- Date value is 0 for 1 Jan, 1 for 2 Jan ..... 31 for 1 Feb

* Eg: print(0) = "1 Jan", print(31) = "1 Feb", ...

fractional marks: subtype of integer

- Can use another basic type for real numbers - called Float
  → But our values are going to typically be only upto 2 decimal places (e.g. 75.25
  so, we have to write constraints for the float values.

- What if we just multiply the fractional number by 100?
  → Then the fractional value at most 2 decimal places will become an integer
  → We can do corresponding operations on the integer values (remembering that they have been scaled by 100)
  → And finally we print the output, we scale the number down and print it

Eg:- Store 62.5 as 6250 and let print(6250) = "62.5"

Introduction to complex datatype

Record (also called struct or tuple)

Data type with multiple fields - each of which has a name and a value

- Sanity of field is ensured by data type specification

List

- A sequence of data elements (for example a sequence of records)
- MarkCardList - is the data type for our data set of all marks cards
  → Each element in the sequence is of ~~WordInPara~~ MarksCard Record data type
- ParagraphWordList - is the data type for our word data set.
  → Each element in the sequence is of WordInPara Record data type
- ShoppingBillList - data type for the shopping bill data set.
  → We need to define the record datatype for a shopping bill

↔ ItemList is a list of Item data type

Summary

- A data type defines the values that the variable can take, and the set of operations that are permitted on it
- Basic data types - Integer, Boolean and Character are needed for our datasets
- Subtype of a type can be used to further restrict values and/or operations allowed
- Record type is a collection of named fields, each with same/different data type
- List type is a sequence of items, all of which are usually of the same data type.