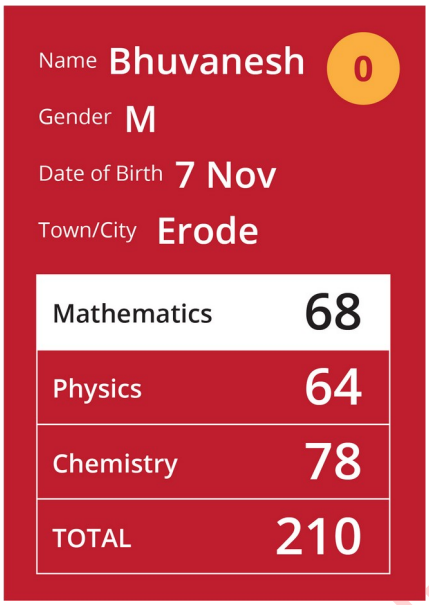
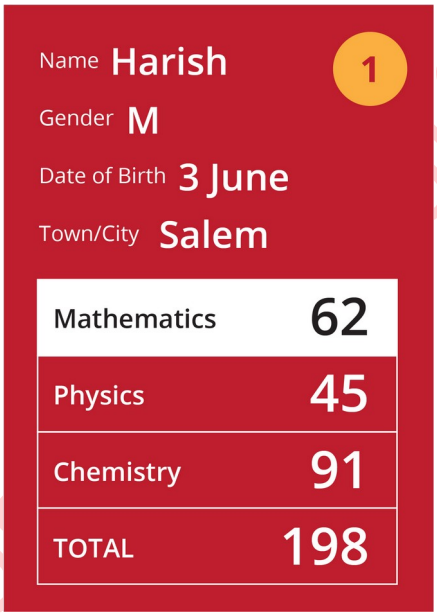
A Brief Summary of week 3

1. **Presentation of datasets in the form of a Table**
   - Each attribute/field is a column in the table (like: Card No., Gender, Total, etc)
   - Each card is a row in the table.
   -

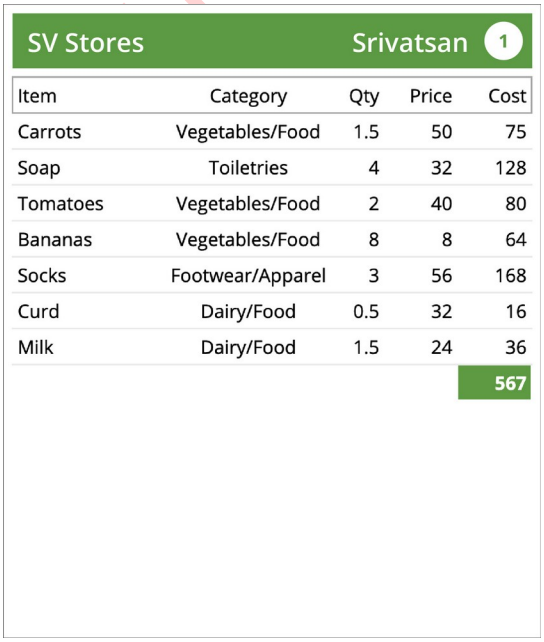
Card No	Name	Gender	DOB	---
0	Bhuvanesh	M	7 Nov	
1	Harish	M	3 June	

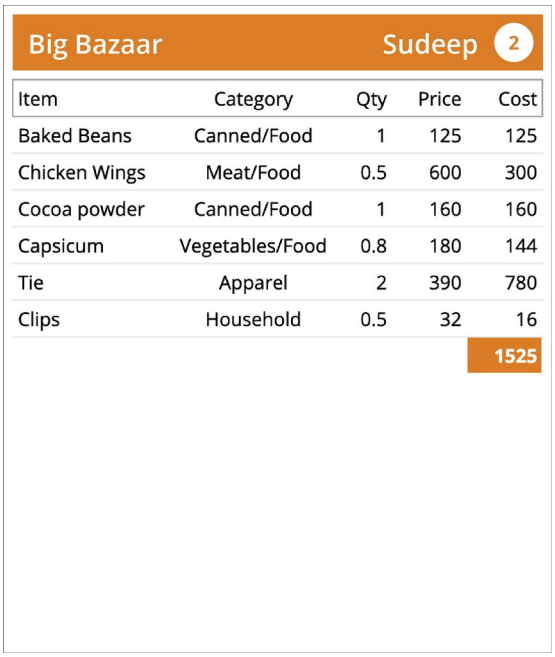
   - Example: Sores dataset, Words dataset.





   - Difficultify when a card has a variable number of attributes
     Multiple rows:– duplication of data
     Split as separate tables:- need to link via unique attributes
   - Example: Shopping bill card





2. **Below average students in two iterations (non-nested) and grade allocation**

- First we compute average marks of some subject (1st iteration)
- And then in next iteration we compare each student marks with average marks of that particular subject

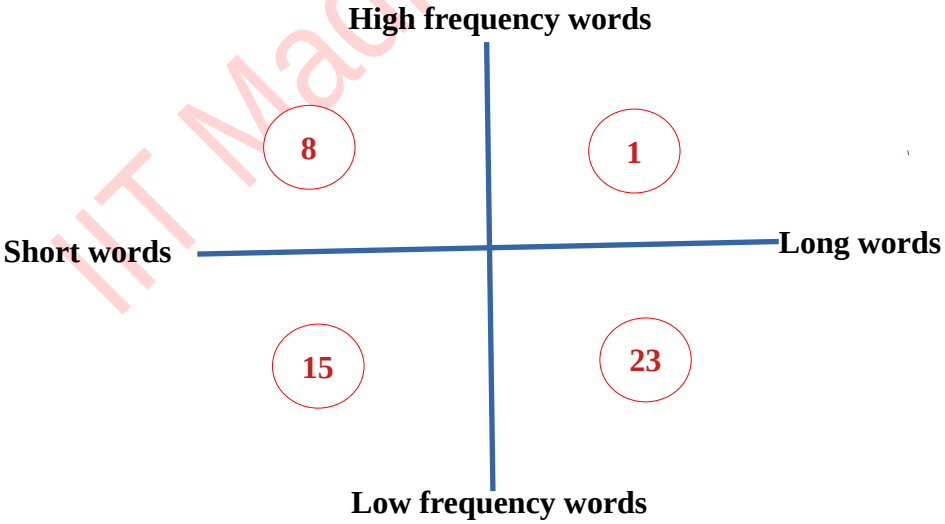
Eg:

```
while(-----){
      sum = sum + X.Maths
      count  = count + 1
}
avg = sum/count
while(-----)
    if(not (X.Maths < avg)) ---- (not True) --- (False)
      avgS = avgS + 1
    }
}
```

Grade allocation: Find maximum and minimum marks of a subject
: Divide into garde bands and
: Count the numbers of students per grade bands

3. **Systematic process of hypothesis verification**

Example: To find the relation between word length and frequency
: Assumption – Long words >= **6** and Short words <= **5**
- High frequency > **1** and Low frequency = **1**



Hypothesis: (i) High frequency words tend to be short.
: (Words which have high frequeny and are short) / (Total high frequency words)
= 8 / (8 + 1) = 8 / 9 = 0.89
(ii) Low frequency words tend to be long?

4. **Three prize problem**

For getting prizes, it should satisfy three conditions:
(i) Should be in top three of total marks
(ii) Should be in a top three in at least one subjects
(iii) And there should be a representation of boys and girls

5. **Introduction to procedures and parameters**

- Procedure name: **SumMaths**
- Argument receives value: **gen**
- Calling procedure with a parameter: **SumMaths (F)**
- Argument variable is assigned parameter value
- Procedure call **SumMaths(F)** starts with, **gen = F**
- Procedure returns the value stored in **Sum** using **return(Sum)**

```
Procedure SumMaths(gen)
    Sum = 0
    while (Pile 1 has more cards) {
      Pick a card X from Pile 1
      Move X to Pile 2
      if (X.Gender == gen) {
        Sum = Sum + X.Maths
      }
    }
    return(Sum)
end SumMaths
```


Interface and Implementation:

Interface: Parameters to be passed
: Value to returned
: What side effects are possible

Implementation: Procedure definition

Eg:

```
Procedure pro1 (a, b)
   return( 0.5*(a+b))
End proc1
```

```
Procedure pro2 ( a, b)
    c = a + b
    c = 0.5*c
   return(c)
End proc2
```

Truth Tables:

(i) AND Table

A	B	**A AND B**
True	True	True
True	False	False
False	True	False
False	False	False

(ii) OR Table

A	B	**A OR B**
True	True	True
True	False	True
False	True	True
False	False	False