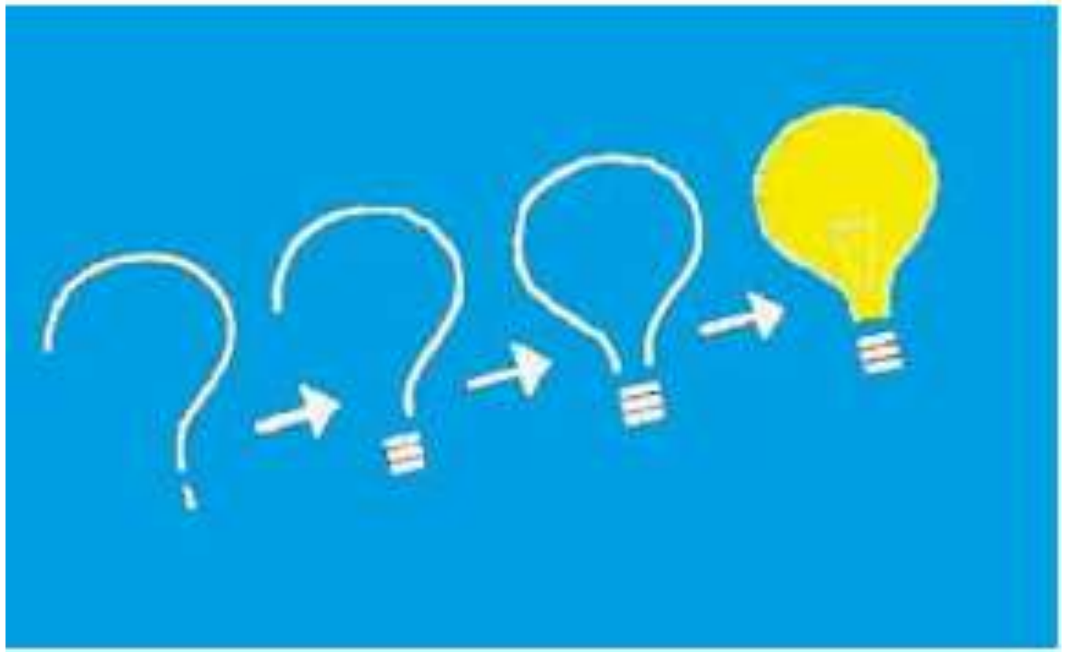
Introduction to Problem Solving

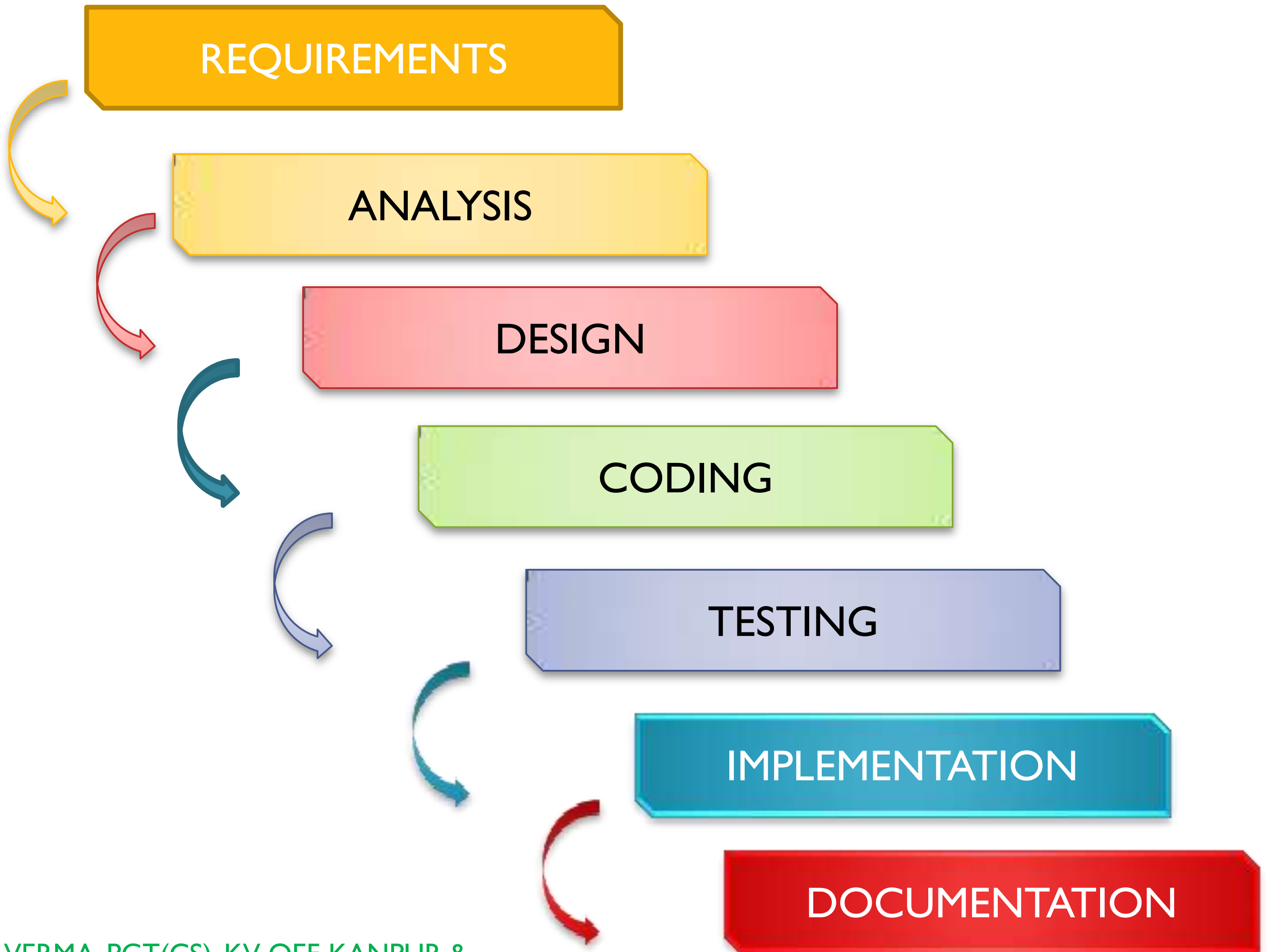


VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHING BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

What is Problem solving?

- Problem solving is a process of transforming the description of a problem into the solution of that problem by using our knowledge of the problem domain and by relying on our ability to select and use appropriate problem-solving Strategies, Techniques and Tools.
- Problem solving (with in the context of developing programs) refers to analyzing a problem with the intention of deriving a solution for the problem

SDLC – Software Development Life Cycle



REQUIREMENTS

- What the problem is?
- What is needed to solve it?
- What the solution should provide
- If there are constraints and special conditions?

ANALYSIS

- **INPUTS:** To the problem, their form and the input media to be used.
- **OUTPUTS:** Expected from the problem, their form and the output media to be used.
- Special constraints or (necessity) conditions (if any).
- Formulas or equations to be used

DESIGN

- In Design phase, we design the proposed solution without writing a computer program by using the techniques:
  - Algorithm
  - Pseudocode
  - Flow Chart

Coding

- It is the process of transforming design (Algorithm, Pseudocode, Flow Chart) in a computer program using any programming language.
- The output of program must be the solution of intended problem.

Testing

- program testing is the process of executing a program to demonstrate its correctness.
- Program verification is the process of ensuring that a program meets user requirement
- After the program is compiled we must run the program and test/verify it with different inputs before the program can be released to the public or other users (or to the instructor of this class)

Implementation

- It is the process of installing the software developed on customer's site.
- Installation may be done by replace the entire manual work with the software developed or in a pilot mode i.e. running the manual and automated system in parallel or installing only a part module with manual work.
- It also involves training of software to the intended users.

Documentation

- Contains details produced at all stages of the program development cycle.

Can be done in 2 ways

    * writing comments between your line of codes.

    * Creating a separate text file to explain the program.

- Important not only for other people to use or modify your program, but also for you to understand your own program after along time.

Algorithm

Watch on YouTube

- An algorithm is a list of steps to be executed with the right order in which these steps should be executed.
- According to D.E. Knuth, an algorithm has the following characteristics:
  - An algorithm has fixed number of steps.
  - Each step in an algorithm specifies the action to be performed.
  - The steps in an algorithm specify basic operations.
  - An algorithm accepts input data.
  - An algorithm generates one or more outputs after the input is processed

Algorithm to withdraw money from ATM

- The algorithm to withdraw a required amount from an ATM will be:
  1. Start the algorithm.
  2. Go to any local ATM.
  3. Insert/swipe your ATM card.
  4. Press the language button that you want to choose.
  5. Enter the pin code number.
  6. Press the account type (Savings or Current) button from which you want to withdraw the money.
  7. Press the cash withdrawal button.
  8. Enter the amount you want to withdraw.
  9. Collect the amount from the ATM machine.
  10. Collect the statement from the ATM machine.
  11. Collect your ATM card.
  12. End the algorithm.

Algorithm of getting out of bed and prepare to go to work

1. Get out of Bed.
2. Take Shower.
3. Get Dressed

If a same steps are performed in a slightly different order:

1. Get out of Bed.
2. Get Dressed.
3. Take Shower

What will be the result?

Just a minute…

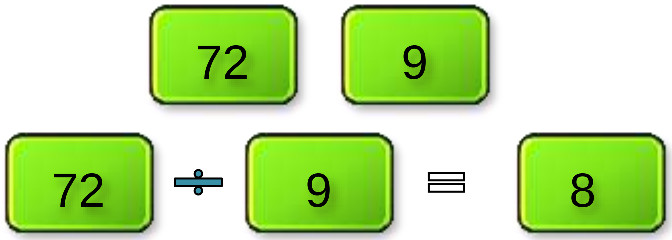
- Write an algorithm to accept two numbers, divide the first number by the second, and display their quotient.
  - *Allocated time to solve the problem is 05 minutes.*

- Write an algorithm that accepts distance in kilometers, converts it into meters, and then displays the result.
  - *Hint: Distance in meters = Distance in kilometers * 1000*
  - *Allocated time to solve the problem is 05 minutes.*

- Write an algorithm that accepts five numbers and displays the sum and average of the numbers.
  - *Allocated time to solve the problem is 05 minutes.*

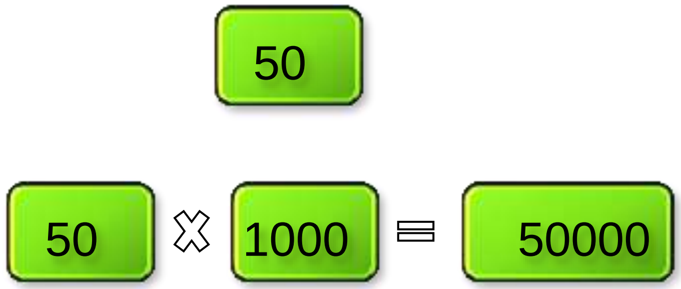
Solution - I

1. Start the algorithm.
2. Get the first number.
3. Get the second number.
4. Divide the first number by the second.
5. Display the quotient.
6. End the algorithm.



Solution - 2

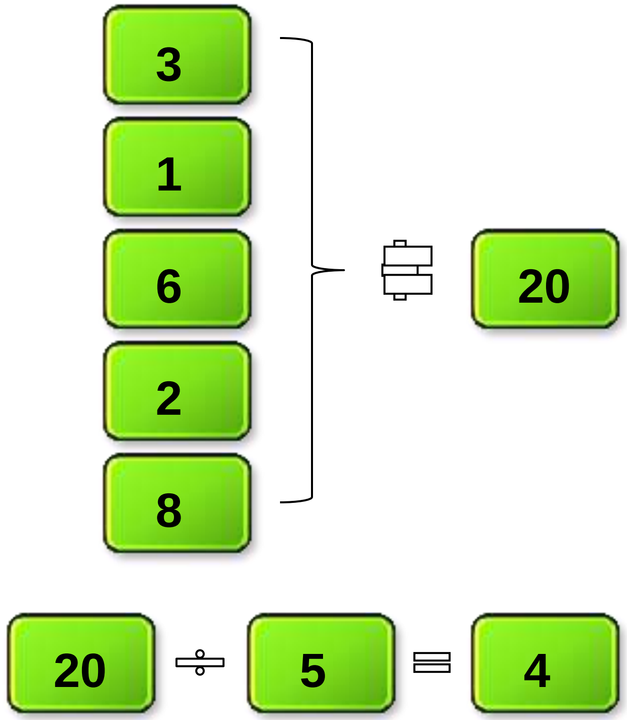
1. Start the algorithm.
2. Get the distance in kilometers.
3. Multiply the distance by 1000.
4. Display the result of multiplication as distance in meters.
5. End the algorithm.



Distance in meters: 50000

Solution - 3

1. Start the algorithm.
2. Get the first number.
3. Get the second number.
4. Get the third number.
5. Get the fourth number.
6. Get the fifth number.
7. Add the five numbers.
8. Divide the sum by 5.
9. Display the addition as sum and result of division as average.
10. End the algorithm.



Sum = 20
Average = 4

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHING BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Flowchart

- Flowchart is a graph used to depict or show a step by step solution using symbols which represent a task.
- The symbols used consist of geometrical shapes that are connected by flow lines.
- It is an alternative to pseudo coding, where as a pseudo code description is verbal. A flowchart is graphical in nature

Advantages & Disadvantages of Flow Chart

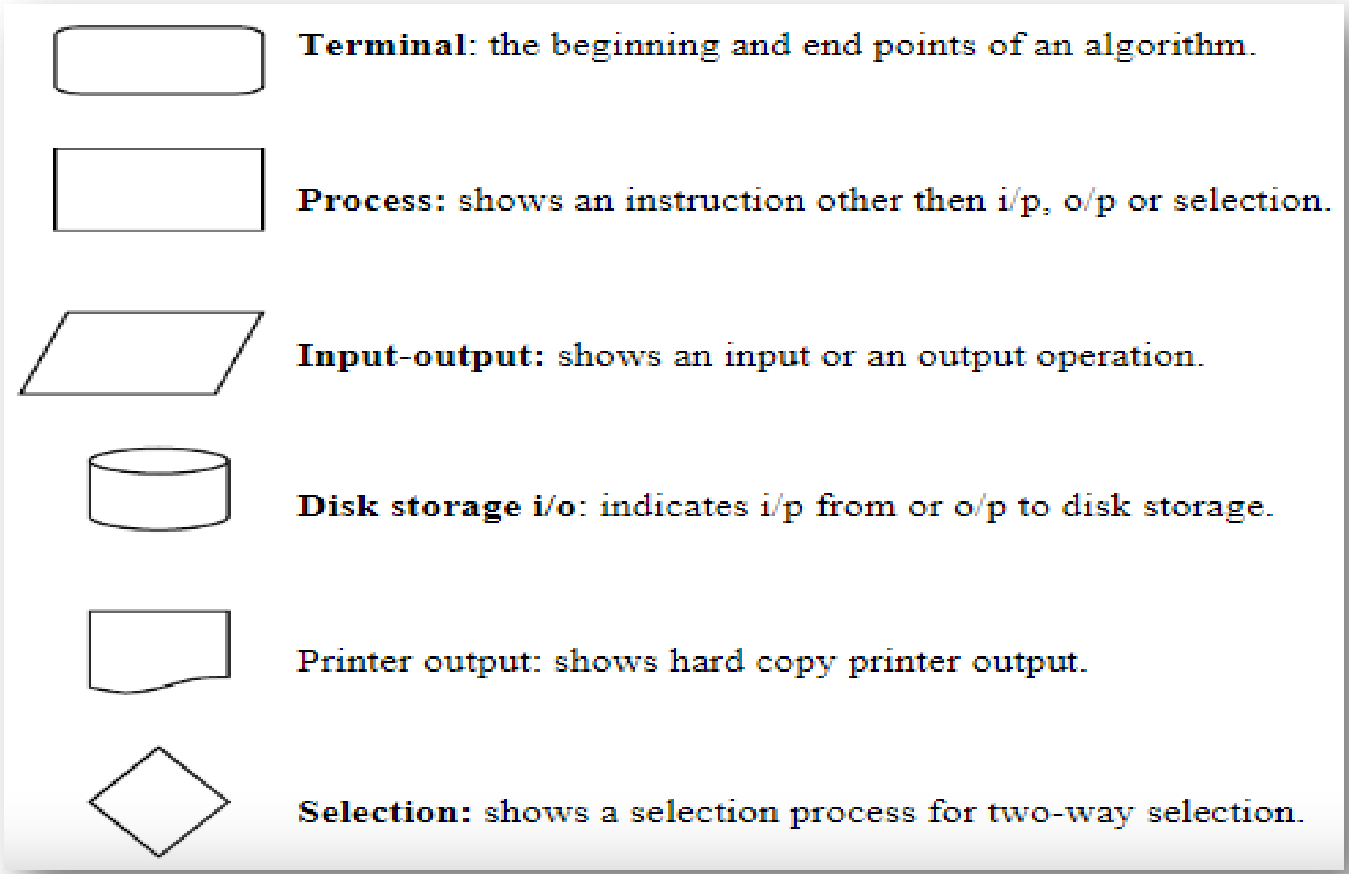
ADVANTAGES

- It helps in analyzing the problems effectively.
- It acts as a guide during the program development phase.
- It help easy debugging of logical errors

DISADVANTAGES

- A lengthy flowchart may extend over multiple pages, which reduces readability.
- Drawing a flowchart is time consuming.
- The changes made to a single step may cause redrawing the entire flowchart

Flowchart Symbols



Terminal: the beginning and end points of an algorithm.

Process: shows an instruction other then i/p, o/p or selection.

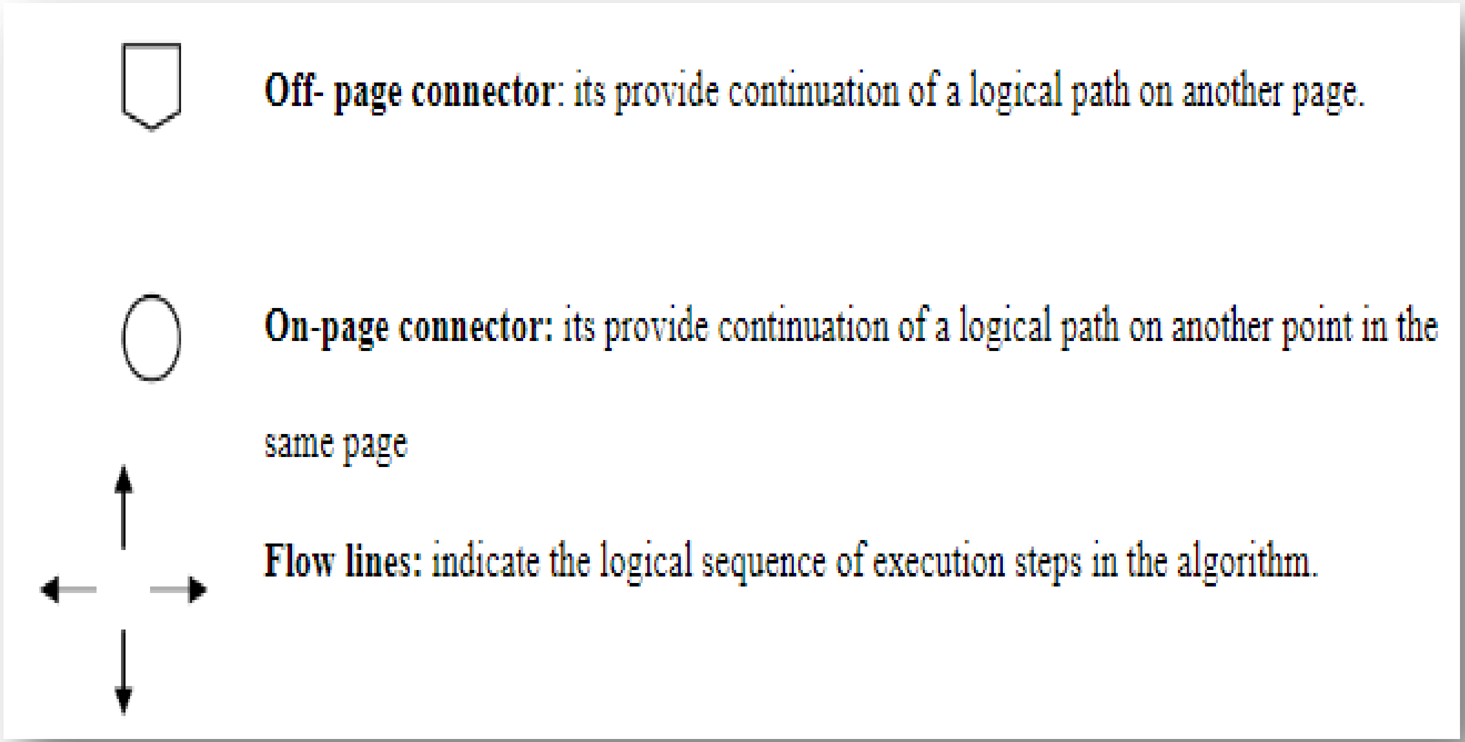
Input-output: shows an input or an output operation.

Disk storage i/o: indicates i/p from or o/p to disk storage.

Printer output: shows hard copy printer output.

Selection: shows a selection process for two-way selection.

Flowchart Symbols



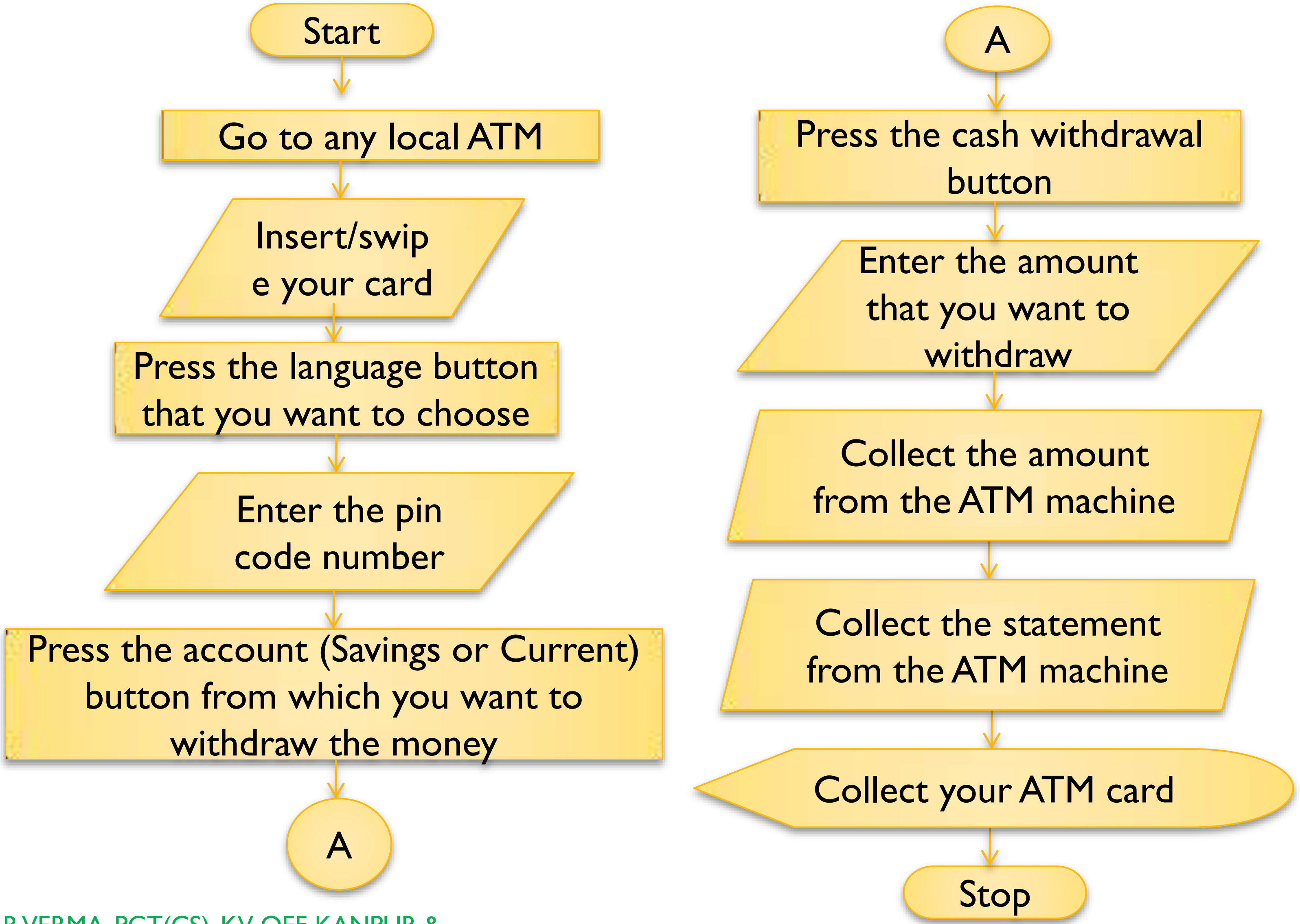
Off- page connector: its provide continuation of a logical path on another page.

On-page connector: its provide continuation of a logical path on another point in the same page

Flow lines: indicate the logical sequence of execution steps in the algorithm.

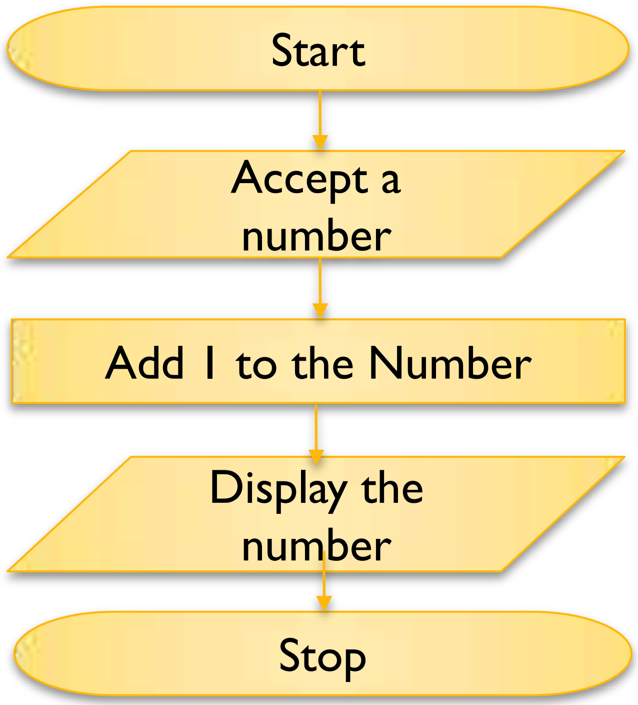
VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHING BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Flowchart – To withdraw money from ATM

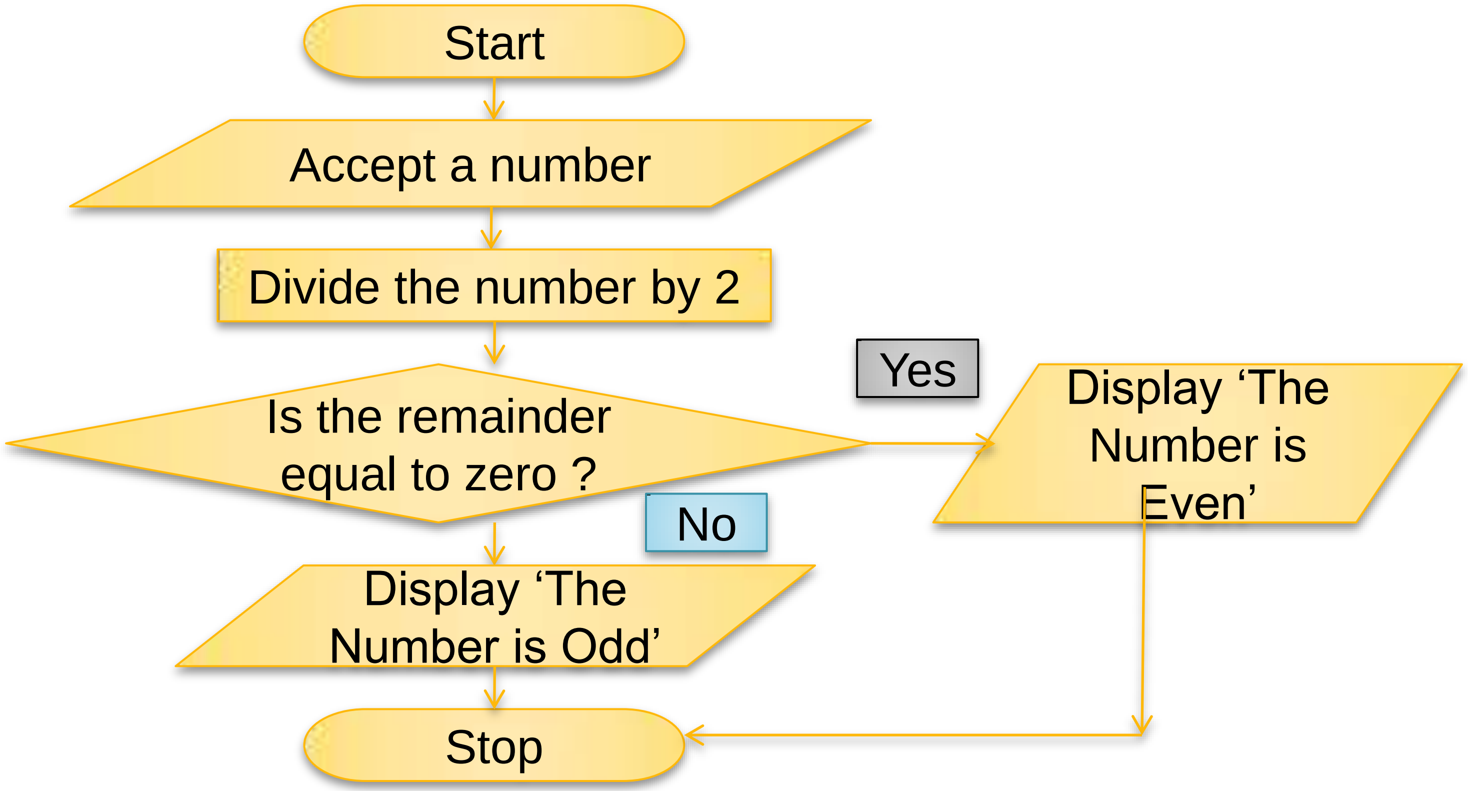


VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHING BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Flowchart – To accept a number and increment by 1, and display the result



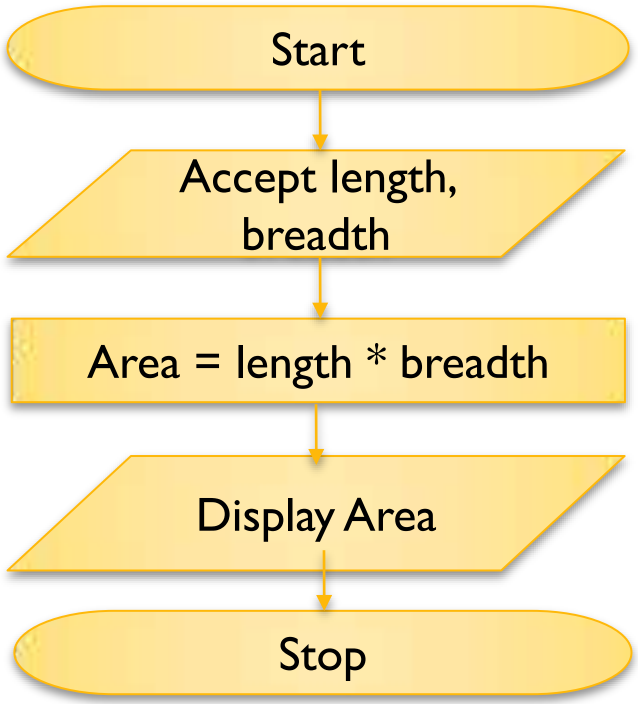
Consider an example where you need to determine whether a number is even or odd. This problem can be solved based on the condition that the number should be divisible by 2. If the number is divisible by 2, it should display the message, 'The Number is Even', otherwise 'The Number is Odd'.


Start
Accept a number
Divide the number by 2
Is the remainder equal to zero ?
Yes
Display 'The Number is Even'
No
Display 'The Number is Odd'
Stop

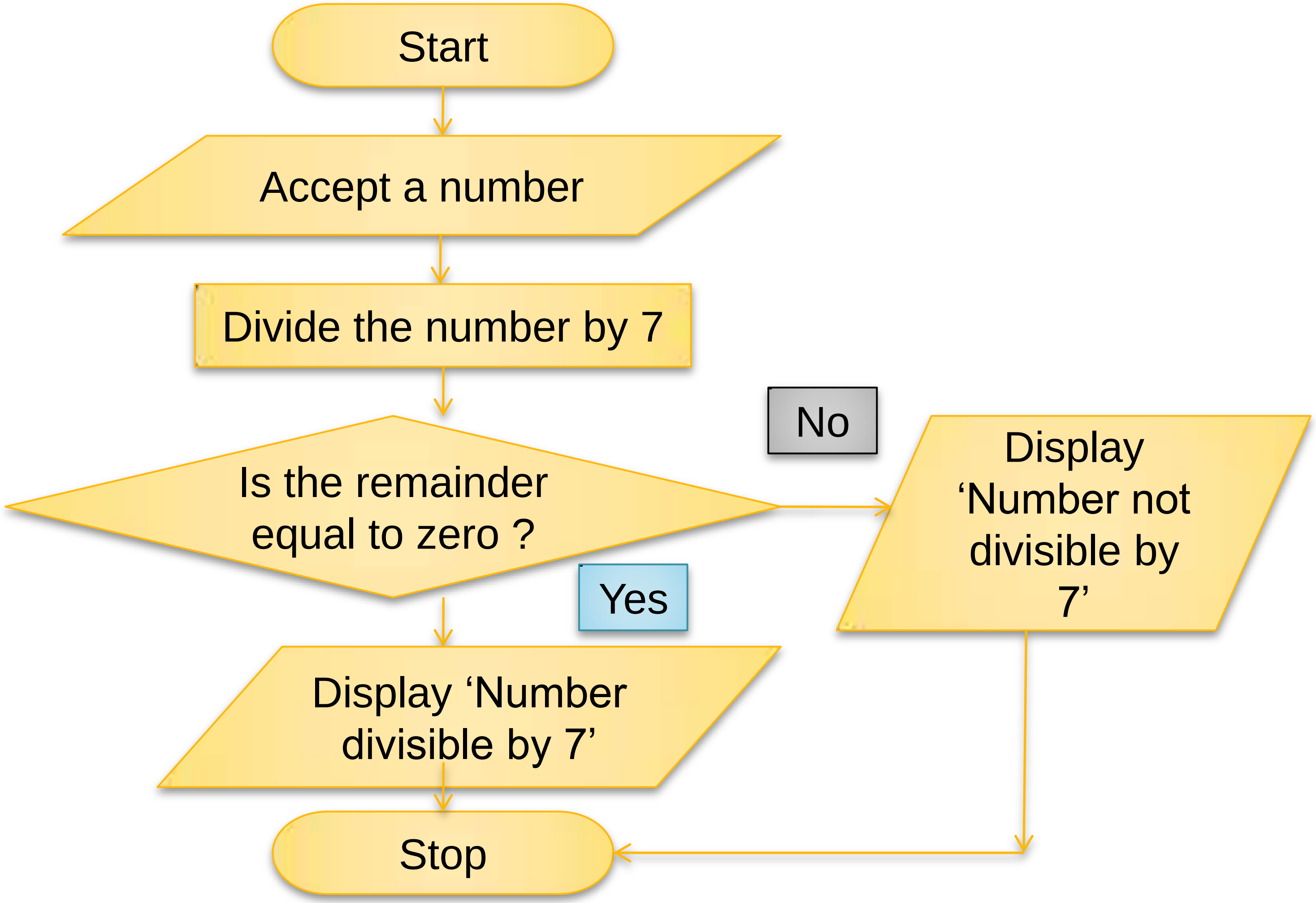
Just a Minute…

- Write a flow chart to calculate area of rectangle
  - *Allotted Time: 05 minutes*
- Write a flow chart to enter any number and check it is divisible by 7 or not
  - *Allotted Time: 05 minutes*

Solution - I


Start
Accept length, breadth
Area = length * breadth
Display Area
Stop

Solution - 2


Start
Accept a number
Divide the number by 7
Is the remainder equal to zero ?
No
Yes
Display 'Number not divisible by 7'
Display 'Number divisible by 7'
Stop

Pseudocode

- It is a formal way of writing the program logic whereas algorithm is an informal way of writing program logic
- It is detailed yet easy description of what algorithm must do.
- It is written in formally styled-English language
- It is used as an initial step in the process of developing of program
- Provide detailed template to programmers.
- It is language independent i.e. it can be converted to program by using any programming language.

Pseudocode

- In Pseudocode we can use few special words for determining the actions like
  - Accept/input
  - Display/print
  - If / else
  - Repeat
  - Start
  - End
  - Goto

Example – to input 2 number and display the sum

1. START
2. INPUT A, B
3. TOTAL = A + B
4. PRINT TOTAL
5. END

Example – to enter a number and check it is even or odd

```
1. START
2. INPUT NUM
3. IF NUM MOD 2 = 0 THEN
4. PRINT "NUMBER IS EVEN"
5. ELSE
6. PRINT "NUMBER IS ODD"
7. END
```

MOD (%) is used to find out remainder by dividing 2

Types of Control program structure

- In computer program there are mainly 3 types of control statements:
  1. Sequential
  2. Selection
  3. Iteration

Sequential

- A series of steps or statements that are executed in a specific order.
- The beginning and end of a block of statements can be optionally marked with the keywords begin & end.
- Example:
    1. Statement1
    2. Statement 2
    3. Statement 3
    4. .
    5. Statement n
- Here each statement will be executed one by one

Example – to swap 2 numbers

```
1. START
2. INPUT A, B
3. C = A
4. A = B
5. B = C
6. PRINT A,B
7. END
```

In this Pseudocode, each instruction will be executed one by one without any jump

Let us input A=10, B = 20
C = A (C = 10)
A = B (A = 20)
B = C (B = 10)

We can clearly observe the values of A and B are exchanged

Can you all think of another method for the same problem without using 'C'
Hint: use arithmetic operation

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHING BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Selection

- It determine two course of action depending upon the condition.
- A condition is given here, which will result in either True(1) or False (0)
- The keyword either if and else (almost all programming language also support this keyword)
- It is also known as branching statement, because it provide two branch, one for True and another for False. Branching will be done depending upon the condition.

Example- to input age and check age is eligible for voting or not

```
1. START
2. INPUT AGE
3. IF AGE>=18 THEN:
4. PRINT "ELIGIBLE FOR VOTING"
5. ELSE
6. PRINT "NOT ELIGIBLE FOR VOTING"
7. END
```

LET US TAKE INPUT IN AGE IS 29, THEN
IF AGE>=18 (TRUE) SO OUTPUT WILL BE "ELIGIBLE FOR VOTING"
NOW TAKE AGE=14
IF AGE>=18 (FALSE) SO "NOT ELIGIBLE FOR VOTING" WILL BE PRINTED

Iteration

- It specifies block of one or more statement that are repeatedly executed till condition remains true.
- Keyword used are :
  - While (supported by almost all programming language also)
  - If with go to (in Pseudocode only)
- Iteration consists of 3 main parts:
  1. *Initial value (from where iteration begins)*
  2. *Condition (upto what condition iteration will continues)*
  3. *Stepping (Increment/Decrement from Initial value to reach to condition)*

Example: To print all number from 1 to 100 (While)

```
1.  START
2.  A = 1                    [INITIAL VALUE]
3.  WHILE A<=100             [CONDITION]
4.      PRINT A
5.      A = A + 1            [INCREMENT]
6.  END
```

Initially **A =1**, Now condition will be checked **A<=100, will be true**, then it will **print 1**, now the value of **A will be increment by 1** i.e. becomes 2, now again condition will be checked, **True**, and this process will be continued, **After printing 100**, value of A will become 101, condition will be evaluated to **False**, and loop will end

Example: To print all number from 1 to 100 (if with go to)

```
1. START
2. A = 1                              [INITIAL VALUE]
3. PRINT A
4. A = A + 1
5. IF A<=100 THEN GO TO STEP 3
6. END
```

Initially **A =1**, It will print 1, after this value of A will be incremented by 1 i.e. 2, Now the condition will be checked, A<=100 (2<=100) will be true so execution will move to step 3 i.e. it will print 2 and again incremented by 1 i.e. 3 and condition be will checked and printed, this process will be condition, when value of A becomes 101, Pseudocode will end

Just a Minute….

1. Write a Pseudocode to convert temperature in Fahrenheit to Celsius
2. Write a Pseudocode to input age and check age is teenager or not
3. Write a Pseudocode to print all the even numbers from 2 to 100
4. Write a Pseudocode to print all the odd numbers from 100 to 1

Decomposition

- Large program can be tackled with "divide and conquer".
- **Decomposition** happens when we break down a **problem** into smaller parts so we can both understand and manage the **problem** better.
- For large, complex **problems** – we can then analyze, solve and design the smaller parts of the **problem** and then put it all together for a solution.

Advantages

- Different people can work on different subprograms.
- Parallelization may be possible
- Maintenance is easier

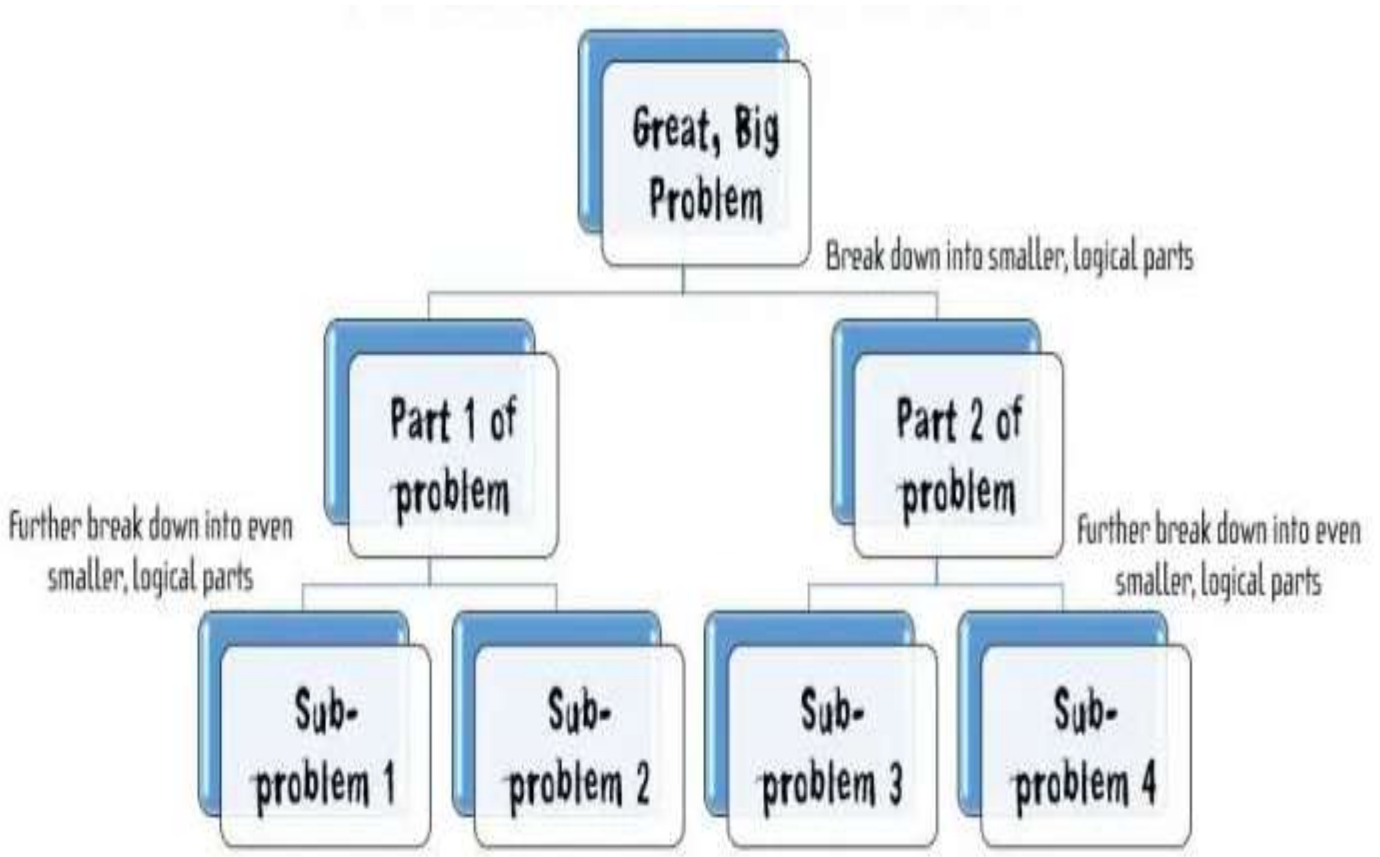
Disadvantages

- The solution to subprogram might not combine to solve the original problem
- Poorly understood problems are hard to decompose

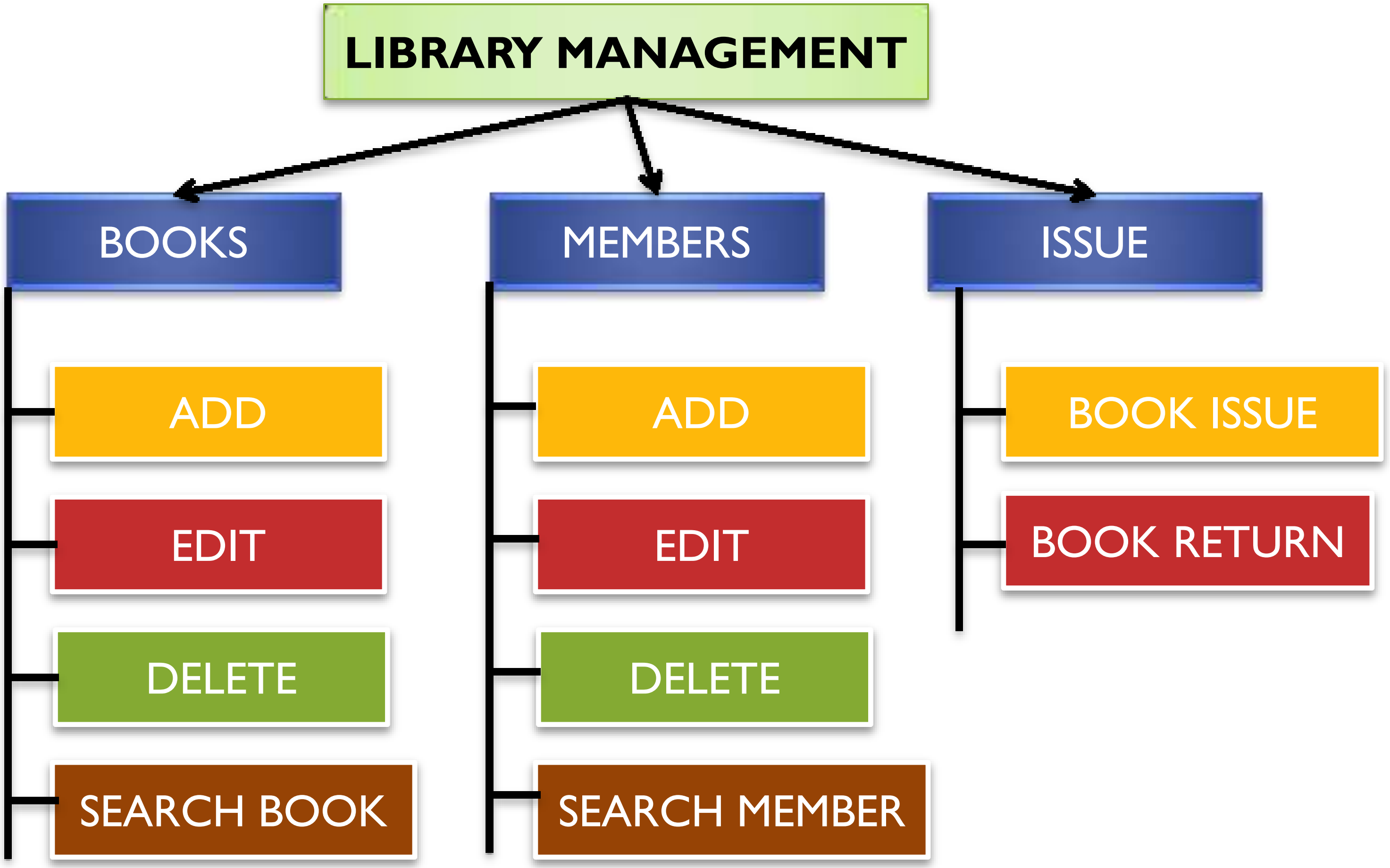
Need of Decomposition

- As studied earlier, first need of decomposition is to divide the large program into smaller and manageable sub program, it is easier to write 2 programs of 400 lines than writing single program of 800 lines.
- Each subprogram can be managed independently
- Global changes can be applied very easily
- Code can be reused
- Functions, modules, class are some example of decomposition.

Decomposition Example - 1


Great, Big Problem
Break down into smaller, logical parts
Part 1 of problem
Part 2 of problem
Further break down into even smaller, logical parts
Further break down into even smaller, logical parts
Sub-problem 1
Sub-problem 2
Sub-problem 3
Sub-problem 4

Decomposition Example - 2



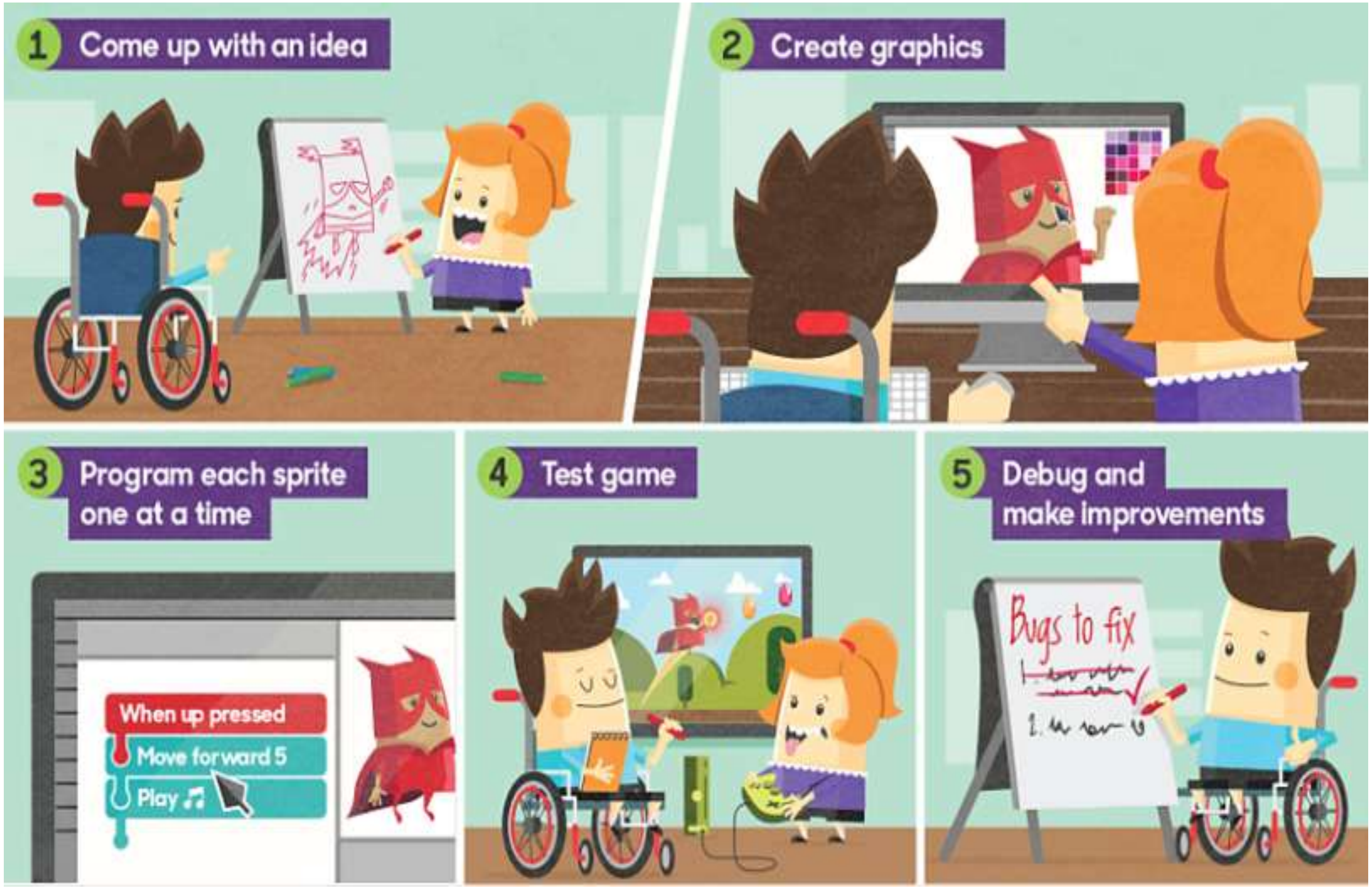
Decomposition Example - 3

- Think of a company creating a mobile phone. Mobile phones are made up of lots of different parts. Companies who make phones might make a list of everything they need and decompose the manufacturing process so that one factory can be making the screens while another makes batteries, another makes the phone case and another makes mother board.

Decomposition – Developing a game

- Game designing is very complex process, it not very simple. Playing game is a simple process but developing a game is not the easy task.
- **Step 1: Came up with an idea to develop a game**
- Step 2: Create all the graphics required.
- **Step 3: Record all the sound needed in game**
- Step 4: Write code for each and every movement (run, jump, sit, walk etc.), action(firing bullets, hit action etc.), score calculation and so on.
- **Step 5: Testing of Game**
- Step 6: Debug and make improvements

Decomposition – Developing a game



VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHING BHARDWAJ, PGT(CS), KV NO.1 TEZPUR