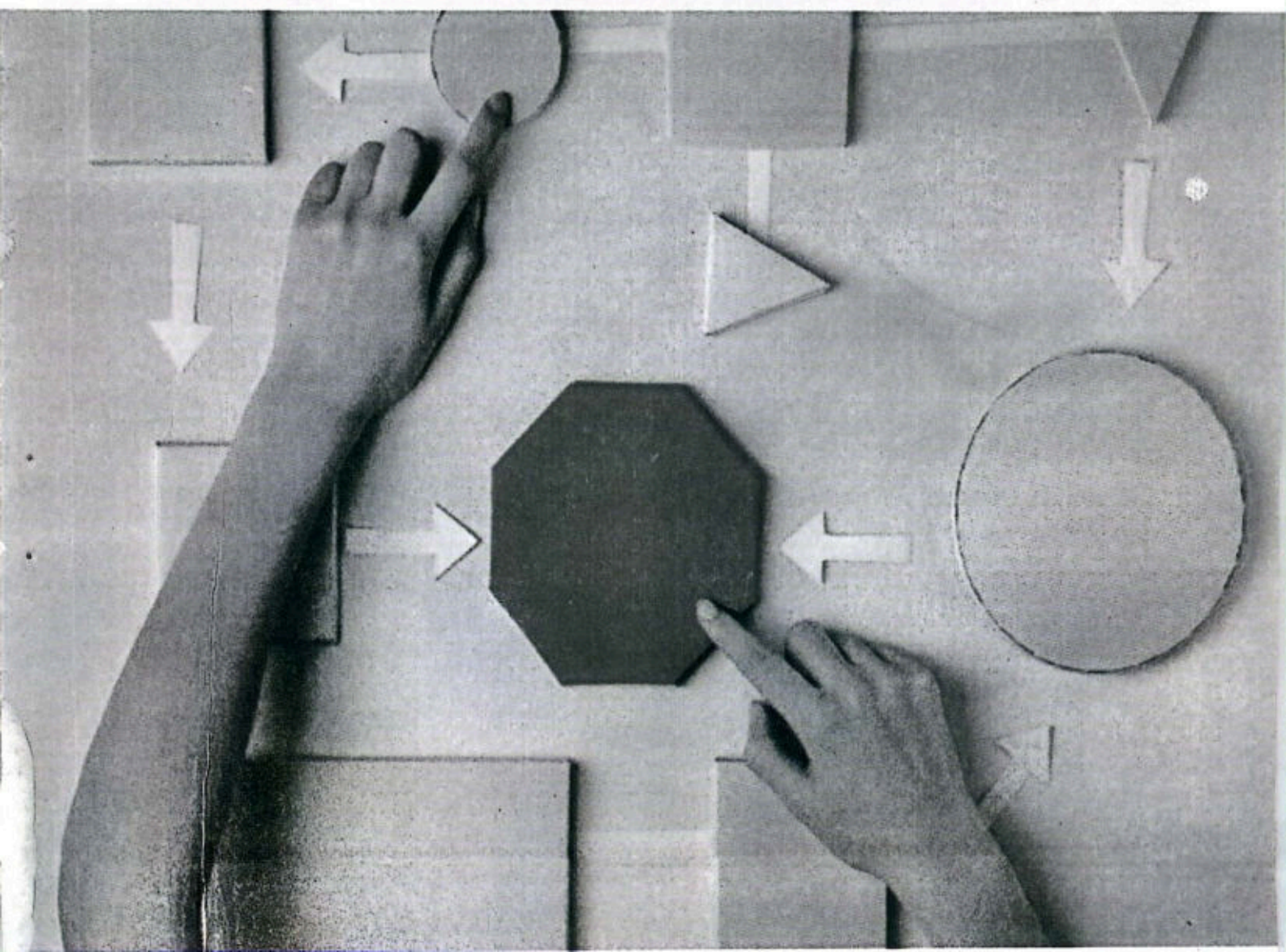
UNIT 2

Computational Thinking & Algorithms

After completing this lesson, you will be able to:

- understand and apply techniques to decompose problems.
- solve simple and complex problems computationally.

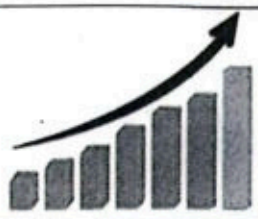


UNIT INTRODUCTION

A computer is a machine that is used to solve problems by accepting inputs, performing operations, and presenting outputs. Computers can't think but can perform those operations that are written by us in the form of algorithms and fed into the computer in the form of programs. You can't program a computer to do something that you don't know how to solve. Therefore, you first need to understand the real-world problem, make some solution, and instruct the computer to behave accordingly when it is asked to solve that problem. Although computers are of various types depending upon their size and capacity but computation is primarily based on processor and main memory. We use computers because they work at a much faster speed and perform operations more accurately than humans.

2.1 Problems

A problem is a challenge or situation that needs to be overcome using some action. The problems are present in all fields such as economic, healthcare, education, transportation, internet, biology and many more.

• What to produce? • How to manage limited budgets?	• How to expand access to medicines? • How to stop infectious diseases?	• How to increase literacy rate in Pakistan? • What subject areas of education should be focused on more?
		
Economic	**Healthcare**	**Education**
• Why is the number of vehicles growing up? • Why the accidents on motorways are increasing?	• How to secure mobile banking transactions? • Why do websites show unwanted advertisements?	• How do 5G cell phone signals harm humans? • How do trees give earth all its oxygen?
		
Transportation	**Internet**	**Biology**

2.2 Identifying a Computing Problem

In computer science a problem is a challenge or situation that needs to be overcome using some action. In computer science, a computing problem is a problem that is solved step-by-step using computation. It can include any type of calculation such as arithmetical or logical. These problems usually have a well-defined input and some desired properties that output must be satisfied. Following are some of the computing problems:

- Decision Problems
- Search Problems
- Counting Problems

Decision Problems: A decision problem occurs when a given input requires a binary response, either Yes or No. Responses may take various forms, such as true or false. Figure 2.1 visually illustrates a simple decision problem. In complex cases, answers extend beyond a straightforward Yes-or-No, involving multiple decision factors and criteria.

Example 1:

- The problem whether a given number is odd (or even).

This is a decision problem where the task is to check if a given integer is divisible only by 1 and itself. If it is, the answer is "Yes," otherwise, it's "No."

Example 2:

- The problem whether a given number is a prime number.

This is a simple decision problem where we check if a number is divisible by 2.The decision is based on whether the number gives a remainder of 0 or 1 when divided by 2.If the remainder is 0, the number is even, and the answer is "Yes".If the remainder is 1, the number is odd, and the answer is "No".

Example 3:

- The problem that is there any occurrence of "aa"?" in a sequence *x* of English alphabets.

The problem is asking for a yes/no answer:

If "aa" is found anywhere in X, the answer is "Yes".

If "aa" is not present in X, the answer is "No"

DO YOU KNOW

Brainstorming is where people sit together with open minds to effectively discuss all possible solutions. The idea is to critically analyze every solution and later choose the optimal one for further analysis and implementation.

Search Problems: In science and engineering, many problems are solved using the search. In such types of problems, we have a set of objects among which we search for the solution. For example, finding a path between two cities. Search problems are often represented using graphs. Where we have nodes and where each link connects to nodes. A node can be connected to multiple nodes. To solve such problems, we have three things:

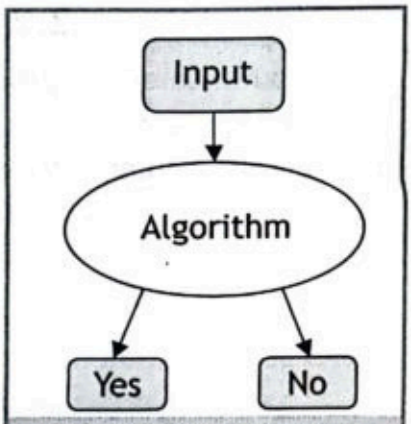


Fig. 2.1: Decision Problem

- *Initial State:* represents the node from where we need to start search.
- *Operations:* represents the moves that transition from one node to another.
- *Goal:* defines the target of end condition.

Example: Route Finding Problem

The map shown in Fig. 2.2 corresponds to the graph, where the nodes are assumed as cities and the links between nodes represent the routes in between the cities. The problem is to find a route in the graph from city S to city T.

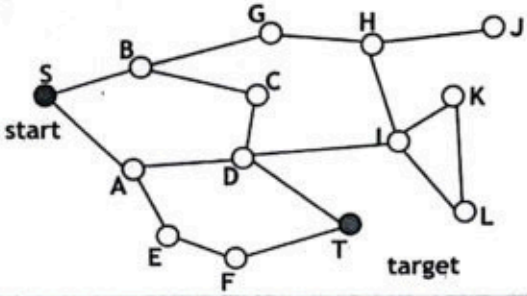


Fig. 2.2: Route Finding Problem

Example: Eight puzzle problem

A 3×3 board is given that contains 8 tiles (every tile can contain one number ranging from 1 to 8) and one empty space. The goal is to place the numbers on tiles in such ordering that it matches the final configuration using the empty space. However, we also have some restrictions that we can only slide one step and only four types of slides are allowed (left, right, above, and below).

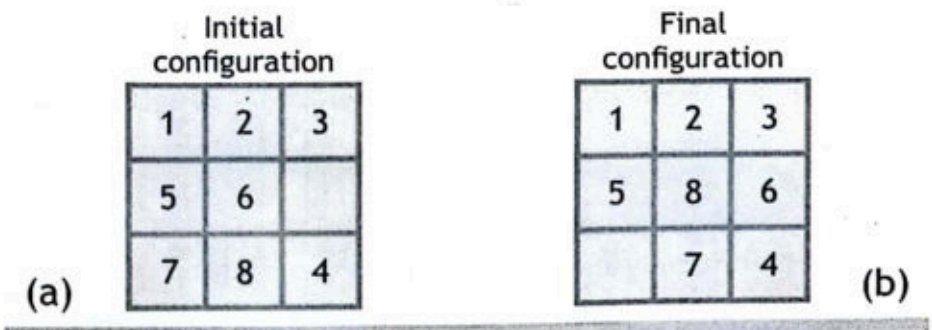


Fig. 2.3: Eight puzzle problem

DO YOU KNOW

Have you used Google Map application?
It is a search problem.

Fig. 2.4: Google Map

Counting Problems: These problems work on the principle that if an event/decision has A number of choices and another decision/event has B number of choices then the total number of possible unique combinations would be A x B.

Let's solve counting problems.

Example: One event is numbers of shirts, and another event is number of pants you own then how many pairs of shirts and pants you can make from it.

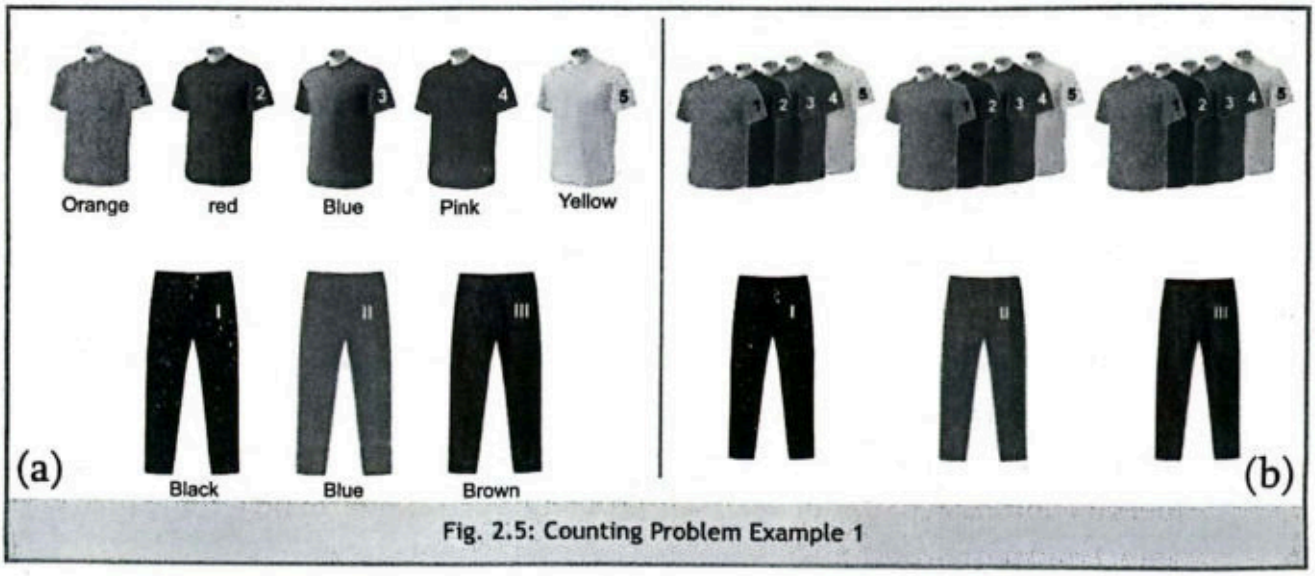


Fig. 2.5: Counting Problem Example 1

Solution

Total possible casual dresses = (number of shirts) x (numbers of pants).

Total casual dresses = 5 x 3 = 15

The counting problems are normally seen as an easy task that can immediately be done. As we go deeper, the counting problems can get complicated very quickly if we want to list out all of the possibilities.

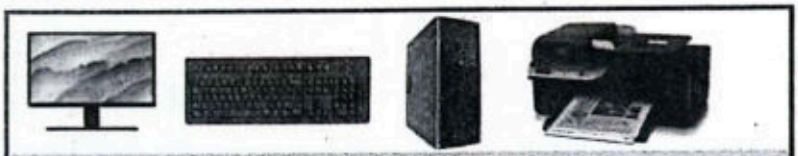


Fig. 2.6: Counting Problem Example 2

Example:

You visit a computer shop to buy a computer system for you. The vender asks you to choose one of 4 monitors, one of 2 keyboards, one of 4 computers and one of 3 printers. How many numbers of possible systems you can choose from?

Solution

You must choose 1 monitor, 1 keyboard, 1 computer and 1 printer. The below given diagram shows each component of the computer system with the number of choices you have.

For the computer system elements, we have 4 monitors, 2 keyboard, 4 computers and 3 printers. Using the counting principle, the number of all possible computer systems that you can buy is given by:

Total possible computer systems = $4 \times 2 \times 4 \times 3 = 96$

2.3 Problem Solving

Problem solving defines the process of analyzing some situation and accordingly behaving to generate some response. For simple problems the following 4 steps are used:

a. Define and analyze a problem: What is the problem and why it is happening.
b. Design a plan: What we are going to do (Algorithm)
c. Implement the plan: Code it by using some programming language.
d. Evaluate: did our plan work?

However, for complex problems, the 6 step problem solving process could be used

a. Define and analyze a problem
b. Decompose the problem: make sub-problems that are manageable
c. Identify potential plans for each sub-problem: such as plan A, plan B and plan C
d. Select and design best plan
e. Implement that plan
f. Evaluate

Before jumping into an action for a problem, it must be thoroughly investigated to capture all the relevant aspects of the problem. For this purpose, we properly define and analyze a problem. In defining a problem, there must be clarity and the objectives of the problem must be specified.

2.4 Input-Processing-Output (I-P-O) Model

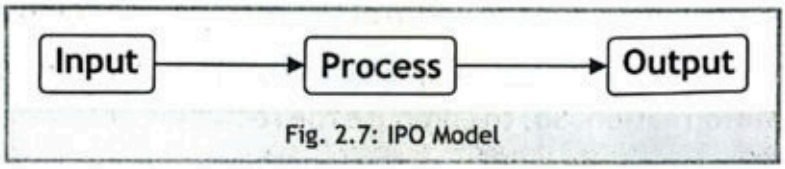


Fig. 2.7: IPO Model

The computer systems work on the input-Process-Output model. The **analyzing of problem allows us** to break the problem down into three components: Input, Output and Processing. it requires us to have correct identification of:

- input that needs to be given to the system,
- operations the system would perform and
- the output to be presented.

Input refers to the requirements from the environment, operations are the computation that is dependent upon the requirement while output is the thing that is presented to the environment.

2.5 Input-Processing-Output (I-P-O) Chart

To visually represent the IPO model, we use IPO charts that represent the Inputs, Process and Output in tabular form. IPO charts are considered a handy tool that software designers use to solve problems.

Input	Process	Output
Data entered in the system	Operations that will be applied on the input	Data that has been turned into the Information

For example, an IPO chart to compute the area of rectangle would be:

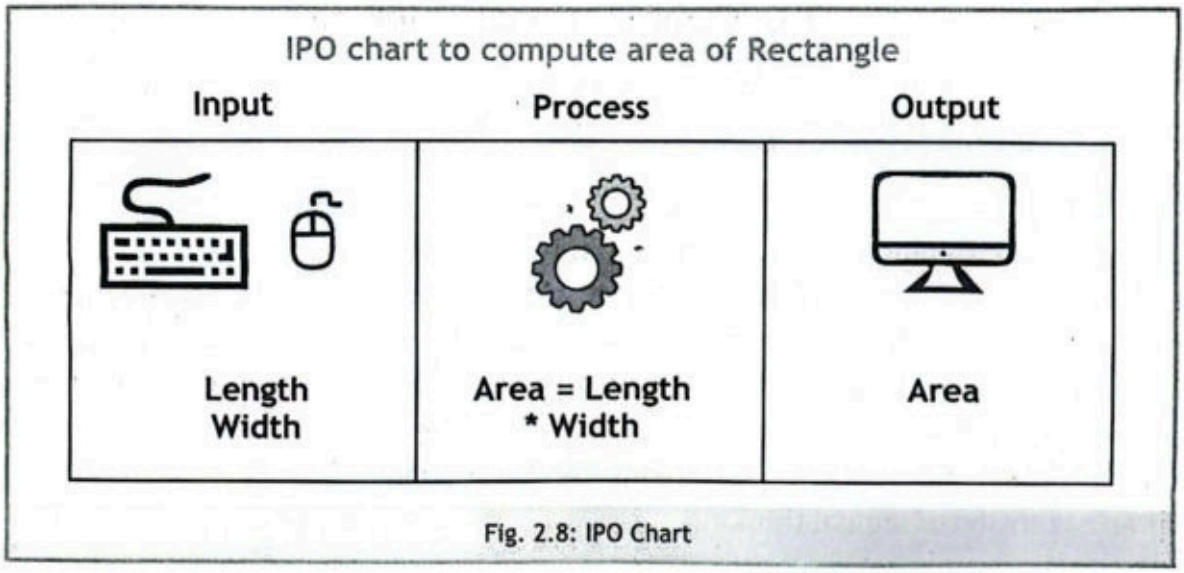


Fig. 2.8: IPO Chart

The input column of the above IPO chart is the input to the system. In this particular example, we need the *length* and the *width* of the rectangle to compute the area.

Process column contains the processing that the system will perform on the input to provide generate information. So, to compute the rectangle area we need to perform the *multiplication operation* on the input e.g. the length and the width of the rectangle.

$$area = length * width$$

The output provides the information to the user. Therefore, the information for the user is the area of the rectangle.

Example

Calculate a student's average based on his grades.

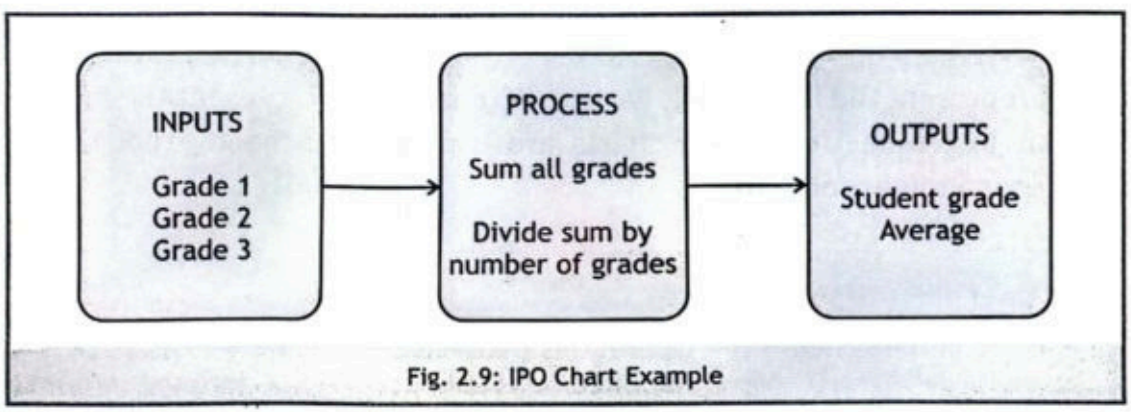


Fig. 2.9: IPO Chart Example

2.6 Computational Thinking

Using computation to solve problems requires the ability to think in a certain way, which is often referred to as 'computational thinking'.

2.6.1 Importance of Computational Thinking

Computational thinking encourages not only to reflect clearly on a problem they're solving but also to define a repeatable solution for it. While solving a problem, we don't just want to solve one instance of a problem, we want an automized system that can solve all instances of similar problem. Computational thinking can be thought of as an extension of logical thinking.

Tip

Before You Learn Programming, Understand Computational Thinking.

2.6.2 Properties of Computational Thinking

- **Decomposition:** Breaking down the larger problems into smaller/ manageable ones and working on them one by one. These smaller problems are referred as sub-problems. This way we simplify the problem and solve it easily.
- **Abstraction:** by removing the unnecessary details to solution, so you could be able to identify essential information.
- **Pattern Recognition:** Examine the problem for a pattern or similarities between previously solved problems.
- **Algorithm Design:** This is actual designing of solution. This involves creating step-by-step plan of the problem solution.

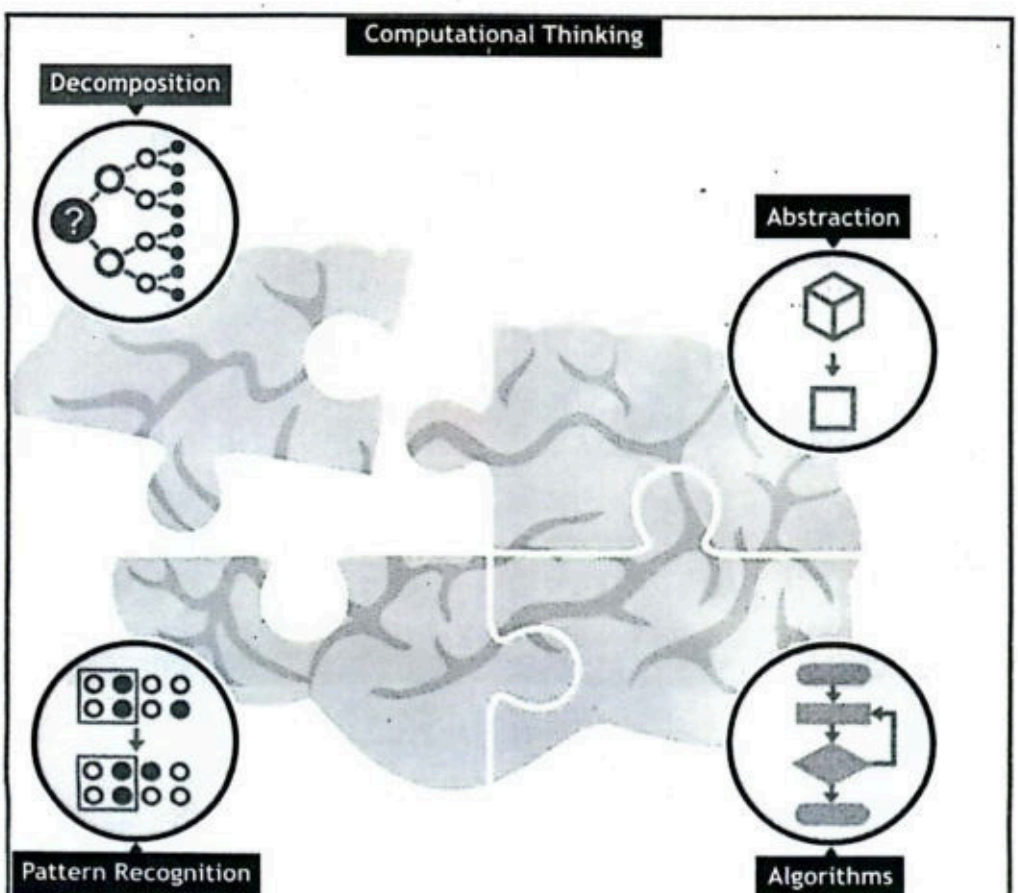


Fig. 2.10: Properties of Computational Thinking

Example: If you want to create your own computer game, how these properties of computational thinking would apply:

Consider the following diagram.

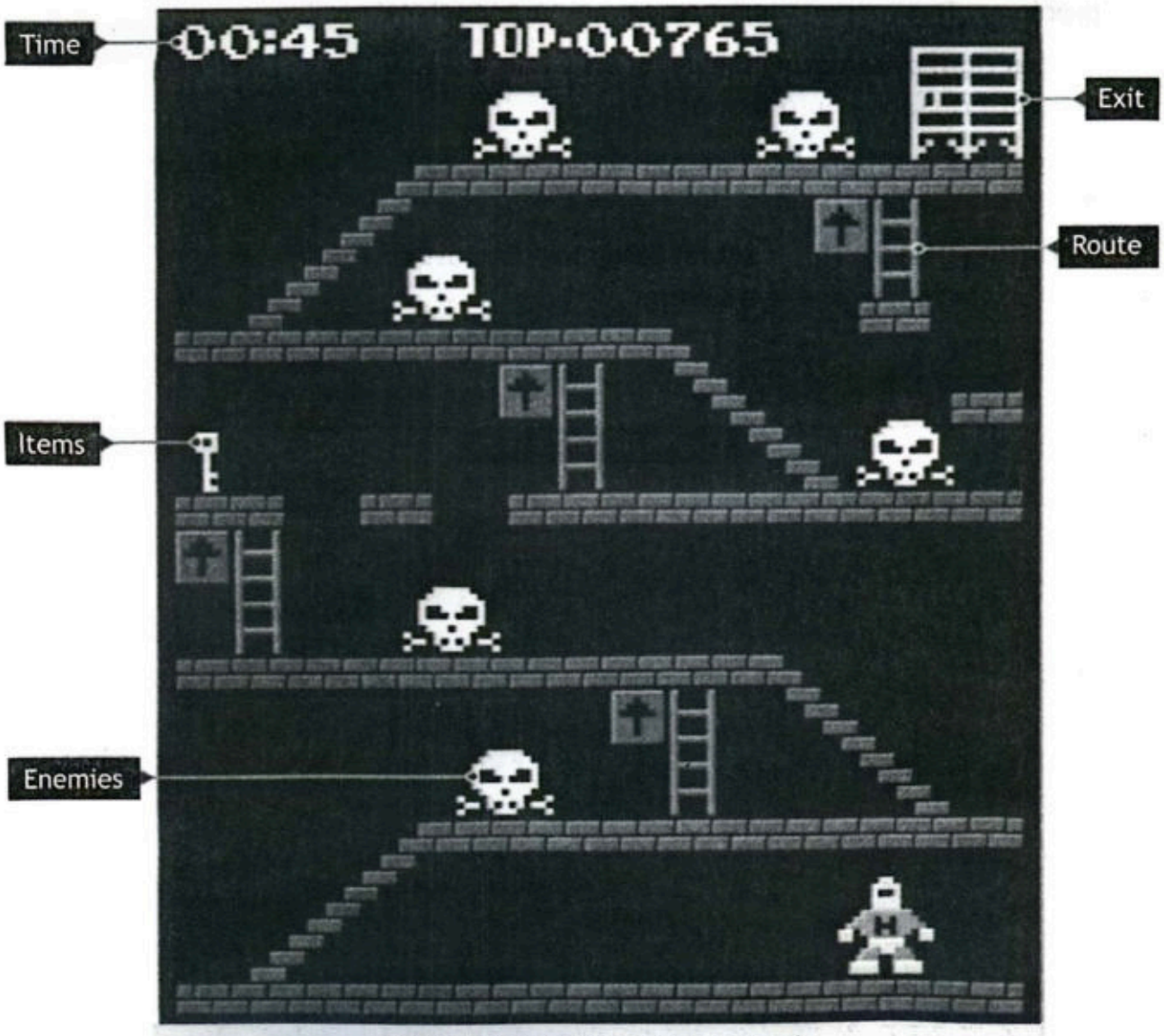


Fig. 2.11: Computational Thinking Example

Decomposition: where to go, how to complete the level

Abstraction:

Necessary Information: location of exit, where is enemy, etc

Unnecessary Information: weather

Pattern recognition: Six enemies should be handled like a single enemy.

Algorithm Design: step by step plan of action e.g. movement

2.7 Principles of Computational Thinking

2.7.1 Logical thinking

Logical thinking refers to analyzing a particular situation or problem using reason and accordingly reaching up to one or more decisions that are sensible. The situation analysis requires gathering of relevant facts and then deciding the best way based on the reasons.

For example, you enter your home and see water on the floor. What do you think has happened?

To solve such problems, you need reasoning skills that are based on evidence and facts.

Example:

Fact: Islamabad is a city
Fact: Cities have a mayor (or some equivalent)
Inferred: Islamabad has a mayor (or some equivalent)

One of the fundamental parts of computational thinking is to think logically. The computers use logic in their computation but it does not mean that they think logically. The computers themselves can't perform logical thinking unless they are programmed to do so. By developing the ability to align problem-solving with technology and focusing on the process as well as the solution

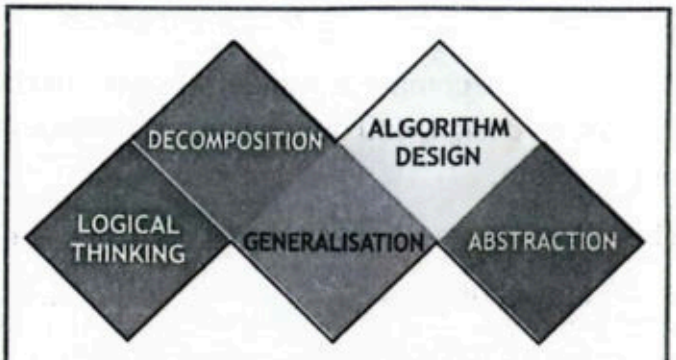


Fig. 2.12: Logical thinking and Computational thinking

2.7.2 Algorithmic Thinking

Algorithmic thinking is about problem-solving. Computers do nothing by magic, algorithmic thinking is a way of getting to a solution. It is a process that involves identifying the steps needed *and then implementing* those steps in a logical and efficient manner. It is a way of thinking that involves breaking down complex problems into smaller, manageable parts and then solving those parts one at a time.

By developing algorithmic thinking skills, you can become a better problem solver.

Example: Find the largest of three unequal numbers.

Let's first draw IPO chart of the problem.

Input	Process	Output
Three numbers	Find largest among the three numbers	Display largest number

Now we write algorithm of the process part of IPO.

Step 1: Let three number be A=10, B=20 and C=30

Step 2: Check if A is the largest?

Step 2.1: Check if A > B and A > C then A is the largest.

Step 3: If A is largest then Stop the process, otherwise proceed to next step.

Step 4: Check if B is largest?

Step 4.1: Check if B > A and B > C then B is the largest.

Step 5: If B is largest then Stop the process, otherwise proceed to next step.

Step 6: C is largest.

2.8 Methods to Design a Solution

Once you have completely understood the problem, the next stage is to design a solution. In solution design, you define how a software will meet the requirements and objectives of a problem.

There are two methods that are used to design a solution:

- Flowcharts
- Concept Maps

Let's discuss both in detail and solve some examples.

2.8.1 Flowcharts

Flowchart is a diagrammatic representation of an algorithm. It describes what operations are required to solve a given problem.

Importance of flowchart in solving a problem

Flowchart illustrates the sequence of operations to be performed to solve a problem in the form of a diagram. Computers programmers draw flowcharts before writing computer programs. It provides an easy way to analyze and find solutions to problems. Once the flowchart is drawn, it becomes very easy to write the program in any computer language. It is very helpful in communicating the problem-solving method to other people. It also helps in finding and removing logical errors.

Steps for drawing flowchart

The flowchart developer must determine the following requirements for the given problem or algorithm before drawing a flowchart.

- Start of the flowchart
- Input to the flowchart
- Type of processing required
- Decision to be taken
- Output of the operation
- End of the flowchart

Start of the flowchart: Every flowchart should start with a trigger. The graphical representation of start of flowchart is oval shape.

Input to the flowchart: The flowchart designer must know what exactly the input to the flowchart is. The input is determined from the problem statement. For example, the given problem is to convert temperature from Fahrenheit to Celsius. Here, the input will be the temperature in Fahrenheit. The graphical representation of input is parallelogram.

Processing in the flowchart: The flowchart designer must decide what type of calculation is to be performed or what formula is applied to obtain the required result. For example, to find the area of a triangle. The following formula is to be used:

Area = (Base x Height) / 2

The graphical representation of processing symbol is rectangle.

Decision making in flowchart: The flowchart designer must decide which control structure (sequence, repetition, or selection) are to be applied for the solution of the problem. For example, selection structure must be applied to print letter grade of a student based on the marks obtained. The selection structure will check in which range the marks fall and accordingly print the grade. Graphically the decision is represented with diamond.

Output: The flowchart must provide the required output. The output is represented with a parallelogram symbol as well.

End of flowchart: The flowchart should have a defined end, and because of the possibility of multiple decision points, it may have multiple ends. For end of flowchart the oval symbol is used.

Flowchart to solve problems:

Flowchart 1: Flowchart to find sum, product, and average of five numbers.

From the statement of the flowchart, it is clear that sum, product and average are to be calculated and given are five numbers. The flowchart is shown below for this.

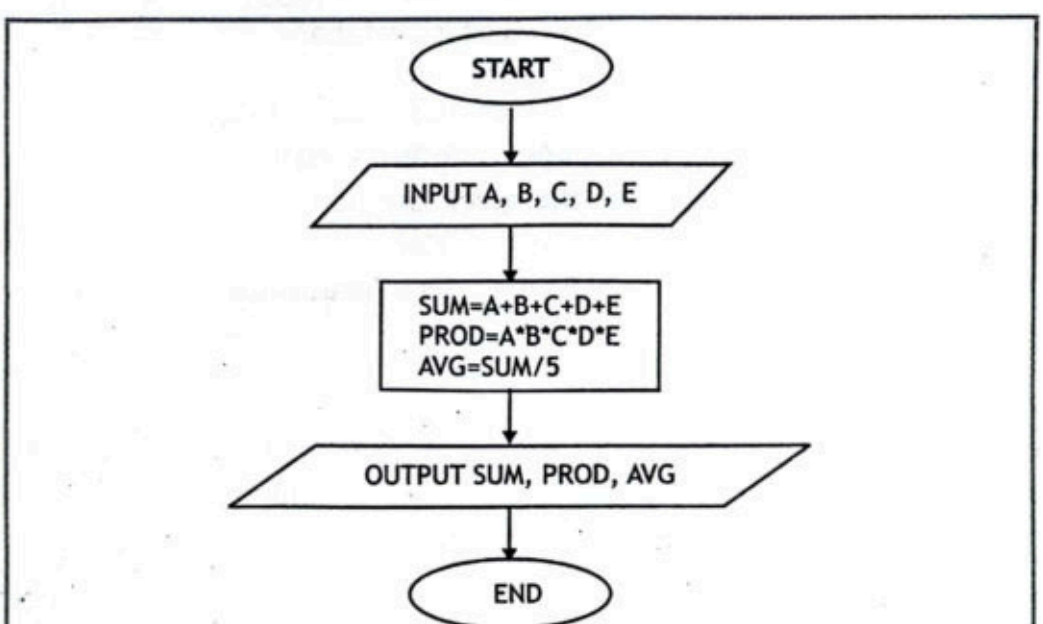


Fig. 2.13: Flowchart to find sum, product, and average

Flowchart 2: Flowchart to find the largest of three unequal numbers.

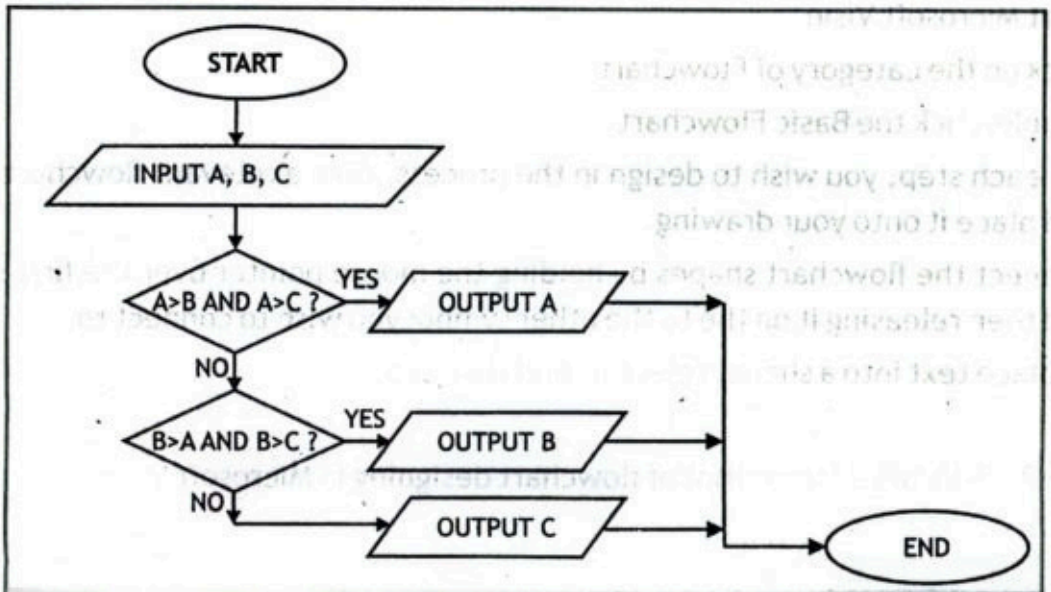


Fig. 2.14: Flowchart to find the largest of three unequal numbers

Flowchart 3: Flowchart to find the area of triangle when the lengths of height and base are given.

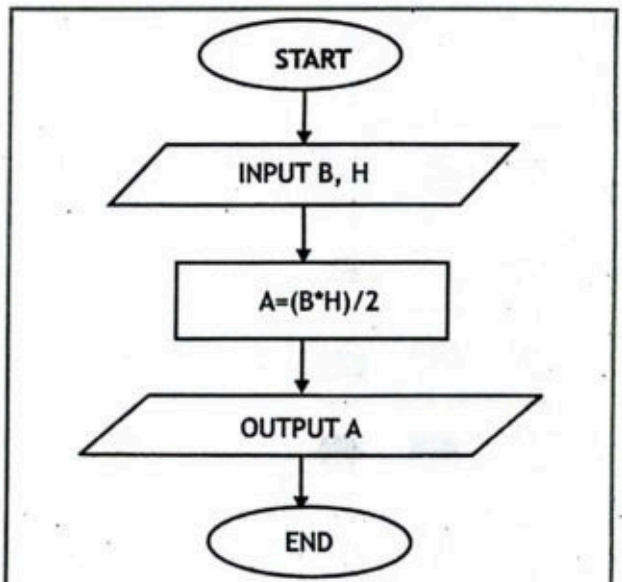


Fig. 2.15: Flowchart to find the area of triangle

Software Tools for Flowchart Designing

For flowchart designing, different software tools are available to design the flowcharts. Some of the famous tools are Microsoft Visio and LARP software.

Microsoft Visio

Microsoft Visio is a tool for drawing various types of diagrams such as flowcharts, building plans, data flow diagrams, network diagrams, etc.

How to create flowchart in Microsoft Visio

a) Start Microsoft Visio
b) Click on the category of Flowchart
c) Double-click the Basic Flowchart
d) For each step, you wish to design in the process, drag a relevant flowchart symbol and place it onto your drawing.
e) Connect the flowchart shapes by holding the mouse pointer over the first symbol, and then releasing it on the to the other symbol you wish to connect to.
f) To place text into a shape, select it, and then type.

The Figure 2.16 shows a screenshot of flowchart designing in Microsoft Visio.

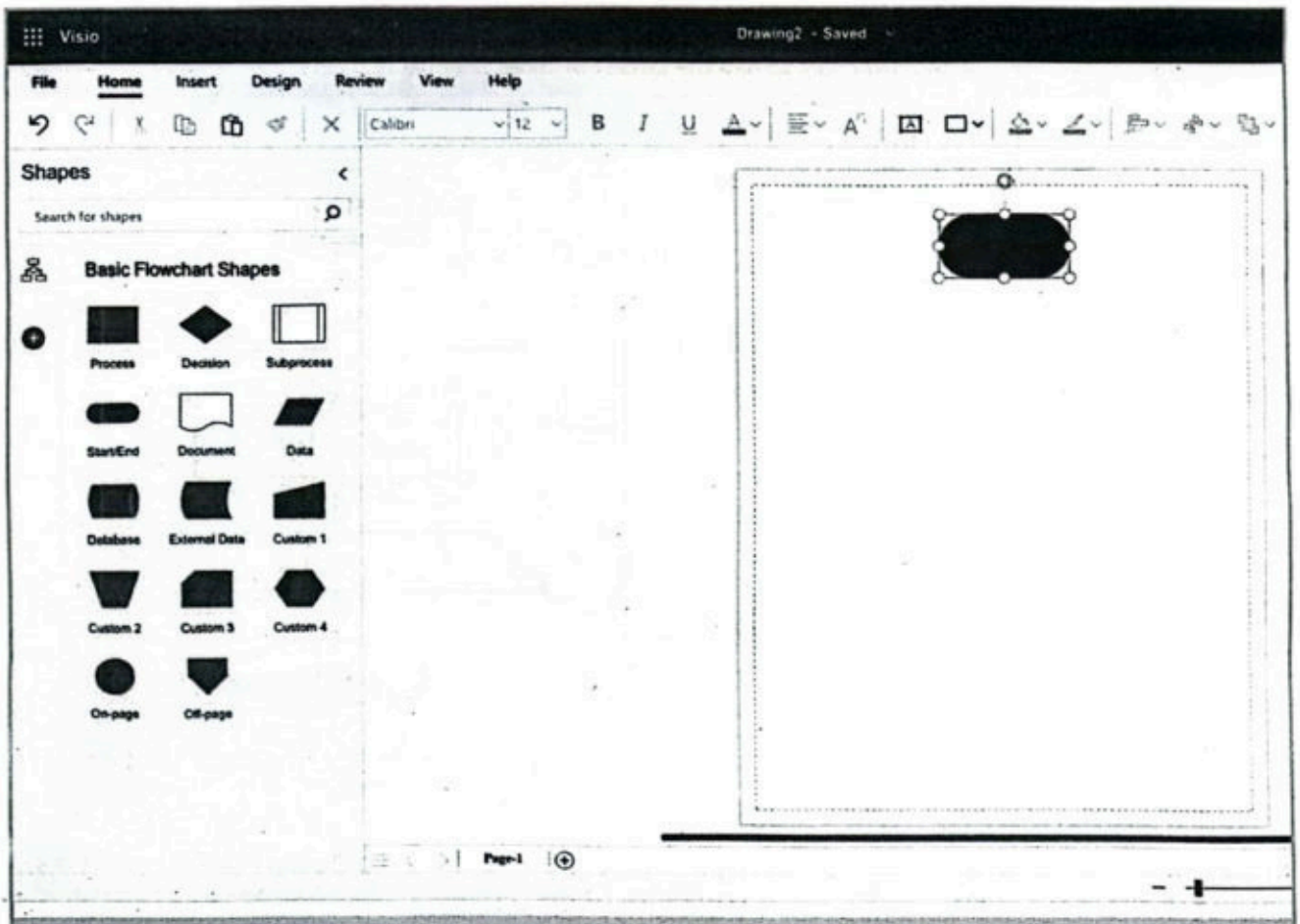


Fig. 2.16: Screenshot of Microsoft Visio for designing flowchart

LARP

Logics of Algorithms and Resolution of Problems (LARP) is a programming language for rapid development of prototypes. However, unlike C++, Java and other programming languages, LARP uses semi-natural syntax. Therefore, it is very easy for non-programmers to understand. LARP also allows algorithms to be expressed as flowcharts. A screenshot of flowchart designing is shown in Figure 2.17.

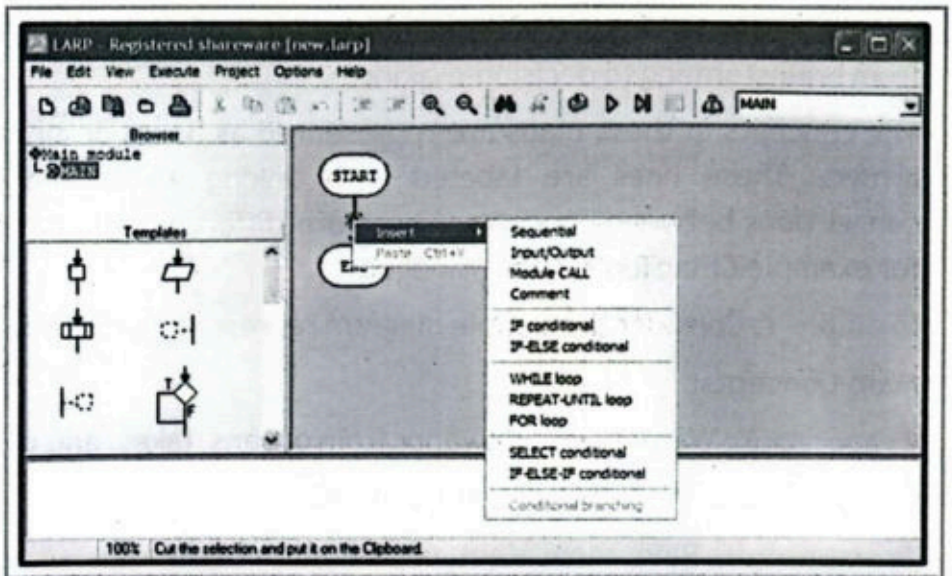


Fig. 2.17: Screenshot of LARP software

2.8.2 Concept Maps

Concept map, like a flowchart, is another way of representing knowledge. A concept map is a graphical tool that represents concepts and relationships between them. This diagram is

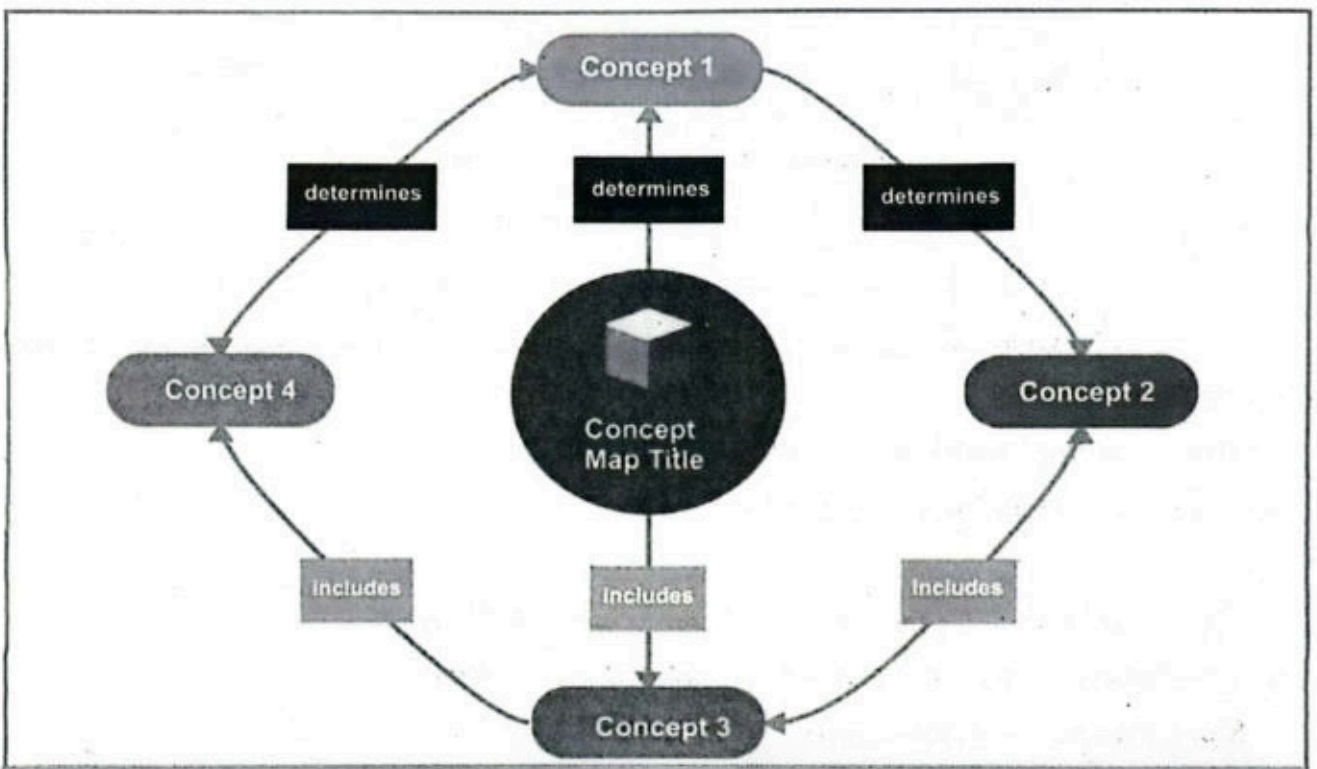


Fig. 2.18: Concept Maps

a popular way to capture understanding of a topic, therefore, it can help with everything from brainstorming to decision-making.

The concepts in these maps are represented as boxes or circles, which relate to lines or arrows. These lines are labeled with linking words and phrases to represent the connections between concepts. There are different software tools for concept mapping, for example CmapTools, Mind Manager.

Example 1: Consider the sample diagram representing concept map of Water Cycle.

Main Concepts:

Evaporation: Water turns into vapor from oceans, lakes, and rivers

Condensation: Water vapor cools down to form clouds

Precipitation: Rain, snow, sleet, or hail fall from clouds

Collection: Water gathers in bodies like rivers, lakes, and oceans

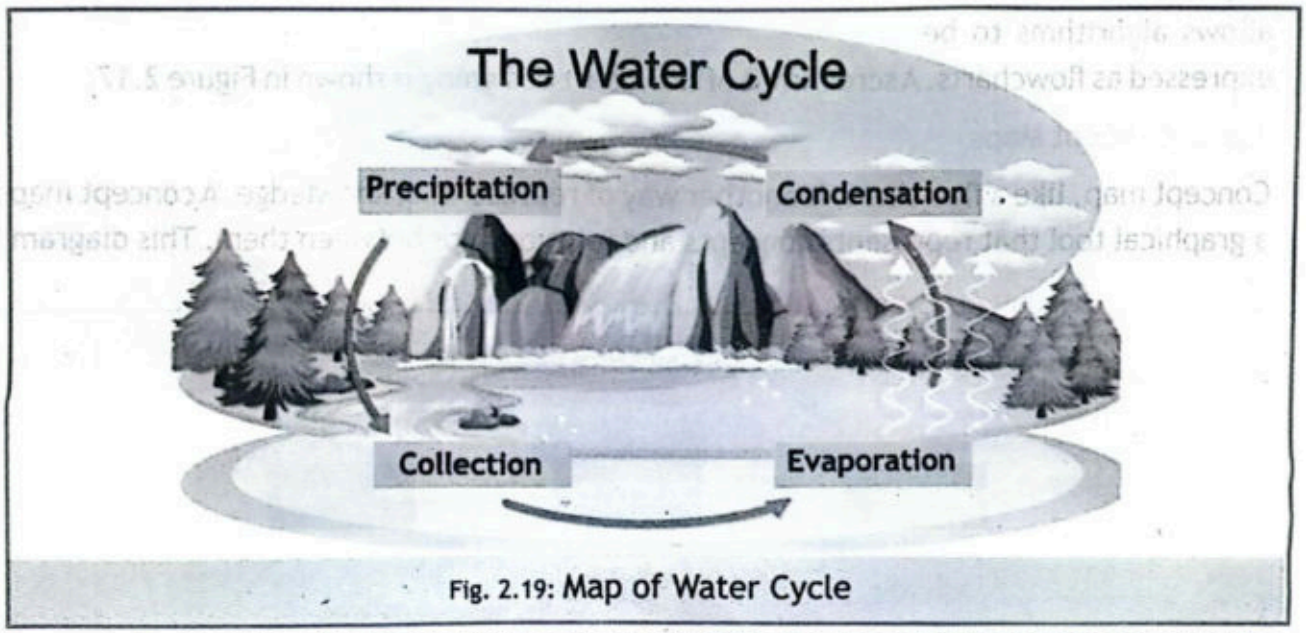


Fig. 2.19: Map of Water Cycle

Example 2: Consider the below given sample diagram representing concept map of Video Game Development.

Concept Map for Video Game Development:

Main Concept: Video Game Development

Game Design

- Gameplay Mechanics (how players interact with the game)
- Level Design (creating game environments and challenges)
- Storyline (plot and characters)

Programming

- Coding (writing the game's code)
- Game Engines (tools like Unity, Unreal Engine)
- Scripting (e.g., how events in the game happen)

Graphics

- 2D Art (sprites, backgrounds)
- 3D Models (characters, environments)
- Animation (movement and transitions)

Sound Design

- Sound Effects (e.g., jumping, background noise)
- Music (background soundtrack)
- Voice Acting (characters' voices)

Testing

- Bug Fixing (finding and solving issues)
- Playtesting (ensuring the game works properly and is fun)
- Quality Assurance (testing for quality and performance)

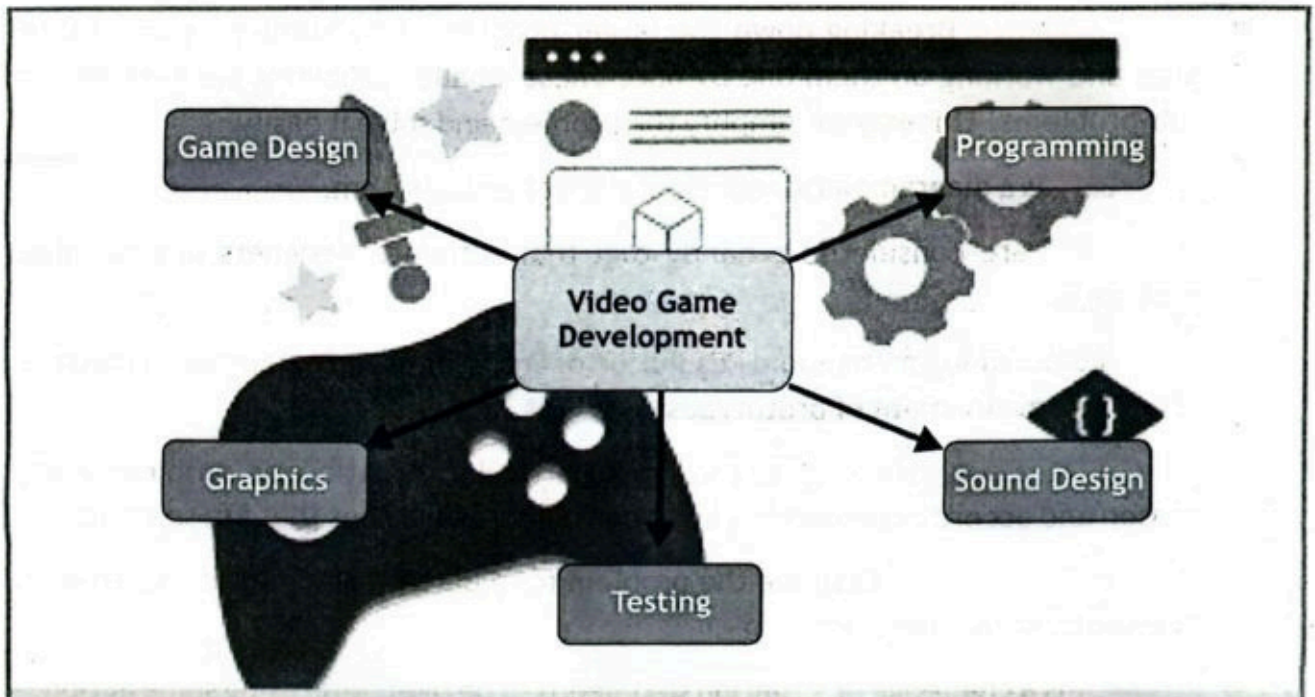


Fig. 2.20: Map of Video Game Development

Summary

- Abstraction: Identifying essential information and Removal of the unnecessary details to solution.
- Algorithm Design: This is actual designing of solution. This involves creating step-by-step plan of the problem solution.
- Algorithmic Thinking is a way of getting to a solution.
- Computing Problem is a problem that is solved step-by-step using computation.
- Computational Thinking is a problem-solving method that computer scientists use as a skill to solve complex problems in a logical and systematic manner.
- Concept Map is another way of representing knowledge.
- Counting Problems: These problems work on the principle that if an event/decision has A number of choices and another decision/event has B number of choices then the total number of possible unique combinations would be A x B.
- Decision Problems: A decision problem is the situation for a given input that has Yes-or-No Answer.
- Decomposition: Breaking down the larger problems into smaller/ manageable ones and working on them one by one. These smaller problems are referred as sub-problems. This way we simplify the problem and solve it easily.
- Flowchart is a diagrammatic representation of an algorithm.
- IPO charts are considered a handy tool that software designers use to solve problems.
- LARP: Logics of Algorithms and Resolution of Problems is a programming language for rapid development of prototypes
- Logical Thinking refers to analyzing a particular situation or problem using reason and accordingly reaching up to one or more decisions that are sensible.
- Pattern Recognition: Examine the problem for a pattern or similarities between previously solved problems.
- Problem is a challenge or situation that needs to be overcome using some action.
- Problem Solving defines the process of analyzing some situation and accordingly behaving to generate some response.

Exercise

Select the suitable answer for the following multiple-choice questions (MCQs).

1. For a problem, we face in real world situations. In what sequence we follow the steps.

   I. make some solution.

   II. understand the real-world problem.

   III. instruct the computer to behave accordingly.

   a) I, II, III b) I, III, II c) II, I, III d) II, III, I

2. Following are types of computing problems

   I. Counting Problems

   II. Search Problems

   III. Decision Problem

   a) I and II b) I and III c) II and III d) I and II and III

3. Computational thinking is

   a) Programming b) Thinking like a computer

   c) Coding d) Logically solving problems

4. To solve Search problems, we need to

   a) Provide the moves. b) Provide start state.

   c) Provide the end state. d) a, b and c

5. The eight queens puzzle is the problem of

   a. Sorting b. Searching

   c. Counting d. Both a and b

6. Finding the location of the element with a given value is

   a) Search b) Traversal

   c) Sort d) None of above

7. In IPO Charts, we have

   a) Input, Plan, Output b) Input, Program, Output

   c) Input, Process, Output d) Input, Proceed, Output

Give Short answers to the following short response questions (SRQs).

1. What is the major difference in solving simple problems and complex problems?
2. Why software designers prefer to use IPO charts?
3. Differentiate between Computational thinking and Logical thinking.
4. Write four properties of Computational thinking.
5. What are the methods used to design a solution?
6. Which Computational thinking technique breaks down the problem into smaller parts?
7. Identify 3 computing problems from other subjects that you are studying in your class.
8. Why do we need to think computationally?
9. The telephone numbers usually have 9 digits. Out of these 9, the first two digits represents the area code and are it remained constant within a given area. The last 7 digits represents the number, and it cannot begin with 0. How many different telephone numbers are possible with a given area code.
10. From city A to city B, there are 4 different roads and from city B to city C there are 2 different roads. Draw a map of given situation and identify how many possible routes are there that someone can follow to reach from city A to city C passing by city B?

Give Long answers to the following extended response questions (ERQs).

Q 1. Identify whether the given problems are Decision Problem, Counting Problem or Search Problem. Write your answer in front of each problem given below:

a. Does a given binary string have an even number of zeros?
b. Flipping a coin result in Head or tails. I flip a coin 20 times, how many different sequences of heads and tails are possible?
c. Does a certain Java program say "yes" to an empty input?
d. How many ways can the letters of the word TRIANGLE be arranged?
e. N-queens problem: where the goal is to place eight queens on a chessboard such that no queen attacks any other.

Q 2. A student has to take one course of physics, one of science and one of mathematics. He may choose one of 3 physics courses (P1, P2, P3), one of 2 science courses (S1, S2) and one of 2 mathematics courses (M1, M2). In how many ways can this student select the 3 courses he has to take?

Q 3. Create an IPO chart which will accept the ages of four boys and calculate their total age and average age. The program must display both the total age and the average age.

Q 4. Create an IPO chart of a scenario that allows a user to enter in two numbers. The operation to be performed is either addition, subtraction, multiplication or division and accordingly the output should be given to the user.

Q 5. The child wants to plan a birthday party for their friend.

a) Draw an IPO chart of this situation.

b) Write down properties on computation thinking.

LAB ACTIVITIES

Activity 1

Draw a flowchart in MS Visio that takes input of two number A and B and output TRUE if A is greater than B otherwise it should output FALSE.

Activity 2

Draw IPO chart and algorithm for the following:

a) Find the exponent of a given number: Exponent or power of a number means how many times to use the number in a multiplication. In other words, it is the product of a number that is multiplied as many times as the exponent.

b) Print odd numbers from 1 to 100. Such as 1 3 5 7 9 11 . . . 99

c) Print the following sequence of numbers in descending order.
27 24 21 18 15 12 9 6 3 0 3 6

d) Find the sum of even numbers up to 100.
SUM = 2 + 4 + 6 + 8 + 10 + 12 + 14 + . . . + 100

e) Print a multiplication table of a given number.

Activity 3

Convert the algorithms of Lab Activity given in Q 2 to flowcharts.

Activity 4

Draw a concept map for different units of computer science books.

Teacher's Guide

A thorough framework for using computational thinking techniques in the classroom is offered by the International Society for Technology in Education (ISTE) Standards. It highlights that educators ought to be capable of "develop algorithms that break down problems into smaller, more manageable subproblems." (https://iste.org/standards/computational-thinking-competencies)