ALGORITHMIC THINKING WITH PYTHON

MODULE - 1

Part II

THE PROBLEM-SOLVING PROCESS: -

Computer as a model of computation,
Understanding the problem,

Formulating a model,

Developing an algorithm,

Writing the program,

Testing the program, and

Evaluating the solution.

Computer as a model of computation

Problem solving can be done by models like

- Conceptual model
- Mathematical model
- Computational model

Conceptual model

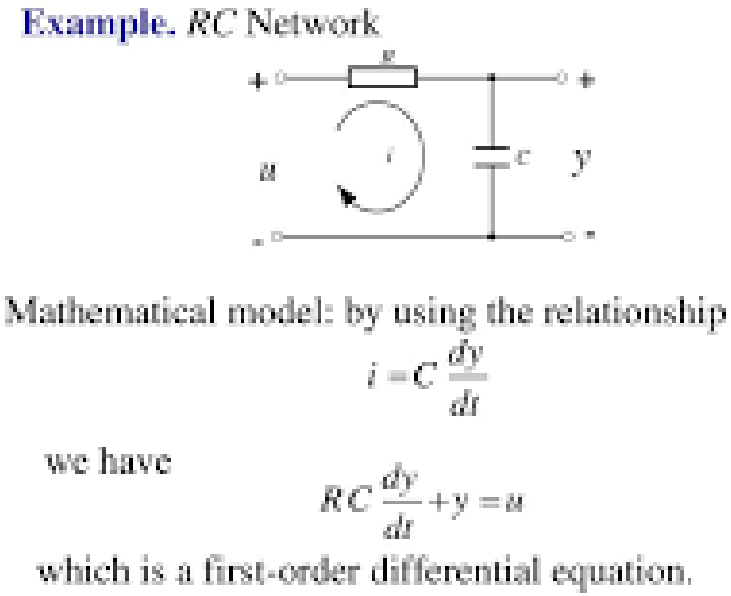
- A conceptual model is a representation of a real-world system or concept that helps people understand it better.
- It is a psychological representations of how tasks should be carried out.

Example: Designing software

Conceptual models can help designers create interfaces and apps that match users' mental models, making it easier for users to learn how to use the product

Mathematical Model

- It uses mathematical expressions to represent the relation between different components of a problem.



Example. *RC* Network

Mathematical model: by using the relationship

$$i = C\frac{dy}{dt}$$

we have

$$RC\frac{dy}{dt} + y = u$$

which is a first-order differential equation.

Computational model

- It is used to simulate the behavior of the solution using algorithm
- In programming, a computer is viewed as a model of computation

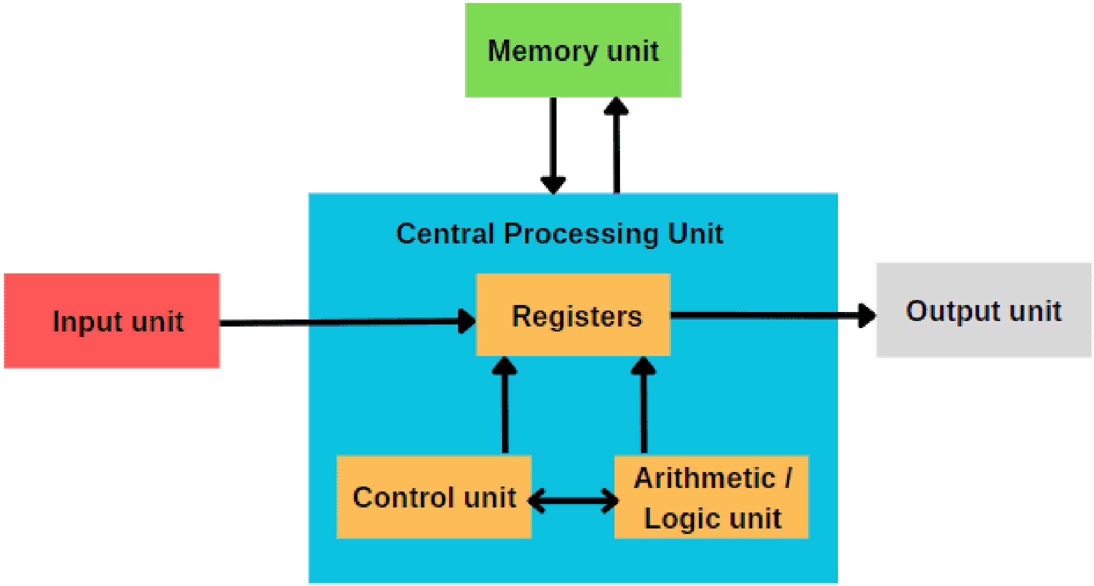


Fig: Basic components of a computer system

1. Understand the problem

- Effective problem-solving demands a thorough knowledge of the problem domain.
- Once you have identified the problem, its exact nature must be sought and defined.
- The problem context, objectives, and constraints if any are to be understood properly.
- Several techniques can be used to gather information about a problem.
- Some of these include conducting interviews and sending questionnaires to the stakeholders

2. Formulate a model

- After the problem is understood, the next step is to devise a solution
- Brainstorming session for ideas
- Develop the models to ensure that it accurately reflects the conceived ideas.
- The model should serve as the blueprint for the algorithm

3. Develop an algorithm

- Models have to be translated into formal representations – *algorithms*.
- Assess the pros and cons of each algorithm to select the best one.
- The assessment is based on considering various factors such as memory, time, and lines of code.

4. Writing the program

- Implement the algorithm as an executable program by writing the codes.
- The program or the code is a set of instructions that is more or less, a concrete representation of the algorithm in some programming language.

5. Testing the program

- Inspect the code to verify its correctness. This is called *testing*.
- During testing, the program is evaluated as to whether it produces the desired output.
- Any unexpected output is an *error*.
- The program should be executed with different sets of inputs to detect errors.
- Debugging of errors is done

6. Evaluating the solution

- This final step is crucial to ensure that the program effectively addresses the problem and attains the desired objectives.
- Define the evaluation criteria. These could include metrics like efficiency, feasibility, and scalability, a few to mention.