B1.1.1 Problem Specification

Overview

Every computer program begins with a **problem to solve**. Before writing code or designing an algorithm, it is essential to clearly define *what the problem is*, *who it affects*, and *what the desired outcome should be*. This process is called **problem specification**.

It ensures that the solution developed meets the needs of its **stakeholders**—the people or groups impacted by the problem or benefiting from the solution.

This topic introduces the foundation of **computational thinking**, which involves breaking down problems logically and systematically so that a computer can solve them efficiently.

1. Significance of Problem Specification

A **problem specification** provides a *clear roadmap* for solving a problem. Without it, developers may waste time building incorrect or inefficient solutions.

Importance:

- Ensures **clarity** — helps identify what the real problem is.
- Helps determine **constraints and limitations** before development.
- Defines **objectives and success criteria** for evaluation.
- Ensures that the final product **meets stakeholder needs**.

For example, imagine developing a school attendance system. If the specification does not include how teachers input data or how parents access reports, the system may not serve all users effectively.

2. Components of a Problem Specification

Component	Description	Example
Problem Statement	A short, clear description of the issue to be solved.	"Teachers spend too much time manually marking attendance."
Stakeholders	People or groups affected by or interested in the solution.	Teachers, Students, School Admin, Parents
Constraints & Limitations	Conditions that restrict the solution, such as cost, time, or technology.	Limited budget, outdated computers, school policy
Objectives & Goals	What the solution aims to achieve.	Automate attendance marking and reporting
Input Specifications	Data the system receives.	Student ID, class schedule
Output Specifications	Expected results produced by the system.	Attendance reports, alerts for absentees
Evaluation Criteria	Standards used to judge success.	Accuracy ≥ 95%, response time < 2 seconds

3. Stakeholders and Their Role

A **stakeholder** is anyone involved in or affected by a project's outcome.
They help identify needs, constraints, and priorities.

Examples of stakeholders in various contexts:

- **School System:** Teachers, Students, Parents, Admin Staff

- **Online Store:** Customers, Developers, Delivery Partners
- **Hospital Management System:** Doctors, Nurses, Patients, IT Support

Why stakeholders are important:
They ensure the product addresses *real-world needs*, not just technical requirements.

4. Constraints and Limitations

A well-defined problem must include **constraints** (boundaries) and **limitations** (factors restricting the design).

Type	Description	Example
Technical	Hardware or software limitations	Old computers, limited storage
Economic	Budget or cost limitations	Limited funding for licenses
Legal	Data protection or copyright laws	GDPR compliance
Operational	Availability of skilled workforce	Few trained staff
Time (Schedule)	Deadlines for delivery	Project due in 3 months

Ignoring constraints often leads to delays or system failure.

5. Input and Output Identification

Every computational solution has:

- **Input:** The data that enters the system

- **Process:** How data is handled
- **Output:** The information produced

Input Method	Examples	Advantages	Disadvantages
Manual Entry	Keyboard, Mouse, Touchscreen	Low cost, flexible	Error-prone, slower
Direct Entry	Barcode Scanner, OCR, OMR	Fast, reduces errors	Needs extra devices
Automatic Entry	Sensors (temperature, pressure, light)	High accuracy, real-time	Expensive setup

Output Types:

- **Temporary Output:** Display on monitor (screen output)
- **Permanent Output:** Printouts or reports
- **Mechanical/Electrical Output:** Triggering actuators or signals (e.g., robot arm movement)

Example:
A smart greenhouse system, inputs temperature from sensors, processes it, and outputs commands to actuators to control irrigation or ventilation.

6. Evaluation Criteria

After defining inputs, outputs, and goals, measurable **evaluation criteria** are set to determine if the final product is successful.

Good criteria should be:

- **Specific** – clear and precise.

- **Measurable** – quantifiable results.
- **Relevant** – directly related to problem objectives.
- **Achievable** – realistic to implement.
- **Time-bound** – evaluated within a clear time frame.

Example:
"The attendance system should mark attendance within 10 seconds per class with 99% accuracy."

7. Computational Thinking Connection

Problem specification is a **core step** of computational thinking.
It involves:

- **Decomposition:** Breaking down the problem.
- **Abstraction:** Ignoring unnecessary details to focus on key elements.
- **Pattern Recognition:** Identifying similarities between problems.
- **Algorithm Design:** Developing a clear step-by-step process.

Example: Designing a **global online learning platform** requires abstraction of details like:

- Regional settings (language, currency, date format)
- Legal compliance (GDPR in Europe)
- Cultural adaptation (UI direction, colour meanings)
- Technical constraints (bandwidth, server locations)

This ensures scalability and accessibility while managing diversity across users.

Questions for Revision

1. **Define** the term *problem specification*.
2. **Define** the term *stakeholder*.
3. **State** three key elements of a problem specification.
4. **List** three examples of technical constraints that could limit a system's design.
5. **Explain** why stakeholder consultation is essential when defining a problem.
6. **Describe** the advantages and disadvantages of *manual data entry* compared to *automatic entry*.
7. **Outline** the steps required to identify inputs and outputs in a system.
8. **Discuss** why evaluation criteria should be measurable and relevant.
9. **Explain** how *abstraction* helps simplify a problem during computational thinking.
10. **Evaluate** whether designing a global online platform is more challenging than a local one, giving reasons related to constraints and limitations.