Typecasting & Decision-Making Statements

1. Typecasting

- **Definition:** Converting a variable from one data type to another.
- **Purpose:** To perform operations between different data types and store values in compatible variables.
- **Types of Typecasting:**
- 1. Implicit Typecasting (Type Conversion / Type Promotion) – Done automatically by the compiler, smaller type to larger type.
- 2. Explicit Typecasting (Type Casting) – Done manually by the programmer using syntax: (datatype)value.
- **Examples:**
- Implicit: int x = 5; float y = x;
- Explicit: float a = 9.8; int b = (int)a; // decimal part lost
- **Notes:** May result in data loss when converting larger to smaller types.

2. Decision-Making Statements

- **Definition:** Statements that allow the program to make choices based on conditions.
- **Purpose:** Adds logic and branching to programs so they behave differently under different conditions.
- **Common Types:**
- 1. if statement – Executes a block if condition is true.
- 2. if-else – Executes one block if true, another if false.
- 3. else-if ladder – Checks multiple conditions in sequence.
- 4. nested if – An if inside another if.
- 5. switch – Chooses a block to execute based on the value of a variable.

3. else-if Ladder

- Used to check multiple conditions one by one.
- The first condition that is true will have its block executed, and the rest are skipped.
- Syntax: if (condition1) { ... } else if (condition2) { ... } else { ... }
- Example: Grading system where marks determine grade.
- Best for range-based or logical comparisons.

4. switch Statement

- Used when a single variable can take multiple constant values.
- Switch directly jumps to the matching case, unlike else-if which checks each condition sequentially.
- Syntax: switch(expression) { case value1: ...; break; ... default: ...; }
- Example: Printing day names based on numeric input.
- Best for fixed options like menu selections or constant values.

5. else-if Ladder vs switch Statement

- else-if can use ranges and complex conditions, switch only uses equality checks with constants.
- else-if works with multiple data types (int, float, char, etc.), switch mostly works with int, char, enums, and strings (in some languages).
- switch is cleaner for many constant options, else-if can be better for complex conditions.

Conclusion:

Typecasting ensures data compatibility between different types. Decision-making statements control the flow of a program based on conditions. else-if ladder and switch are two important decision-making constructs, each suitable for different scenarios.