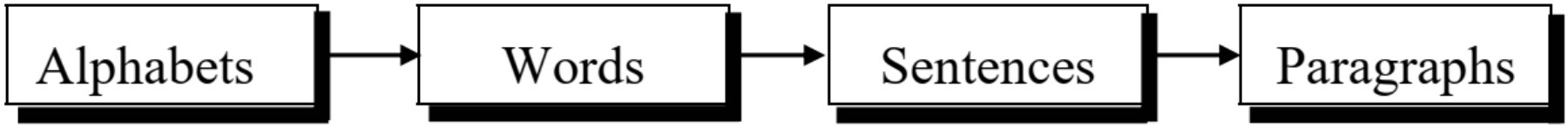
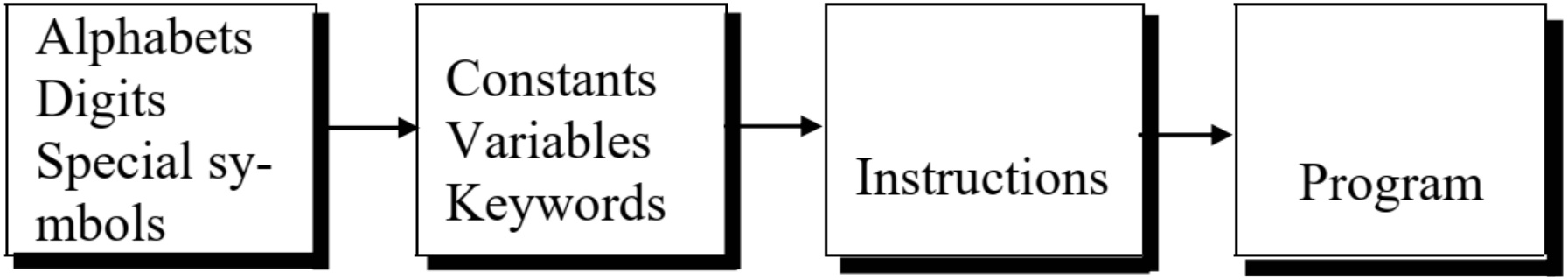
CSN-103: Fundamentals of Object Oriented Programming

Compare English and Programming Language

Steps in learning English language:



Steps in learning a Programming Language



Constants and Variables

- **Constant:** A constant is an entity that doesn't change
  - Example: 23, 5555, A, Rahul, 34.56
  - Also called **Literals in Java**

- **Variable:** A variable is an entity that may change with time
  - Just as in Mathematics
  - Example: x=5, y=2.44

- Why we need variables?
  - For calculations, values stored in the memory (RAM)
  - Memory contains millions of 'cells' or 'locations'
  - Variable names → Points to memory locations → Easy referencing

Java: Data Types and Variables

- Variable declaration and initialization
- Example: Integers
  - Correct way

```
int x;                  // Declaration
int x=10;               // Declaration & Initialization


int x;                  // Declaration
x=10;                   // Initialization
```

  - Incorrect

```
x=10;                   // Initialization without declaration
```

- Declaration for a variable can be done only once

Java: Data Types and Variables

- Java is a strongly typed language
  - Every variable and expression has a type
  - All assignments are checked for **type compatibility**
    - **Automatic conversion** for compatible types
    - No automatic conversion of **conflicting types**

- Strongly vs. Weakly Typed Languages
  - C language is a weakly-typed language
    - int i = 3.0; No Error in C/C++

Java: Data Types and Variables

- Variable names can be a combination of alphabets, numbers, and special characters
- Restrictions
  - Special characters (_ and $)
  - Can’t start with a number
    - 6data : WRONG
  - Can’t have space
    - Roll no: WRONG
- Reserved words (keywords) can’t be used as variable names

Lexical Issues (Vocabulary in Java)

- Keywords: Reserved. Cannot be used as names for variables, methods, and class

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while