

Data Structures



Algorithm, Flowchart

**Subin Sahayam, Assistant Professor,
Department of Computer Science and Engineering
Shiv Nadar University**

Last Class Summary

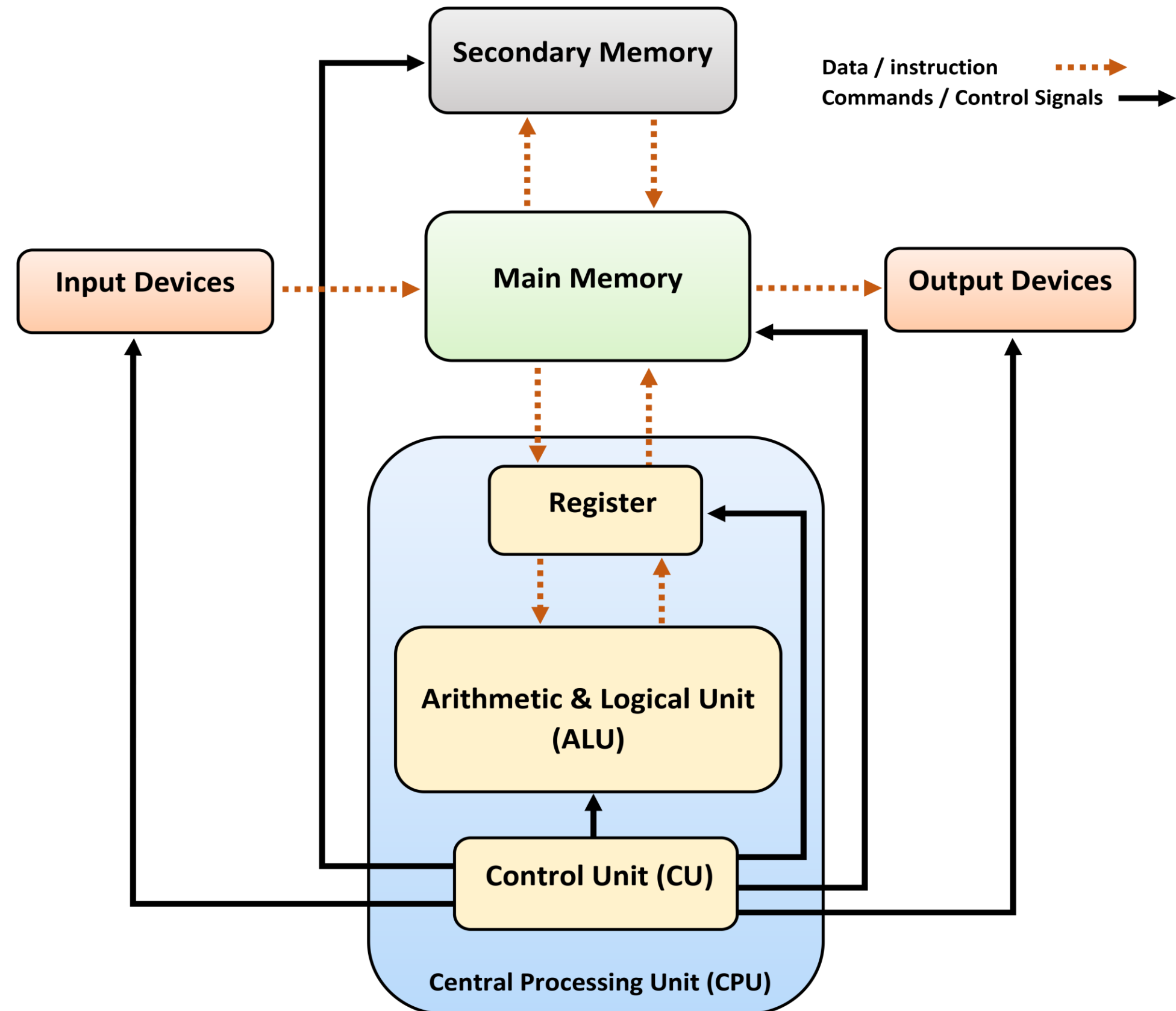
- **Data types**
- **Memory**

Remember

SHIV NADAR
— UNIVERSITY —
CHENNAI

General Parts of a Computer

- **Processor**
- **Memory**
- **Input**
- **Output**



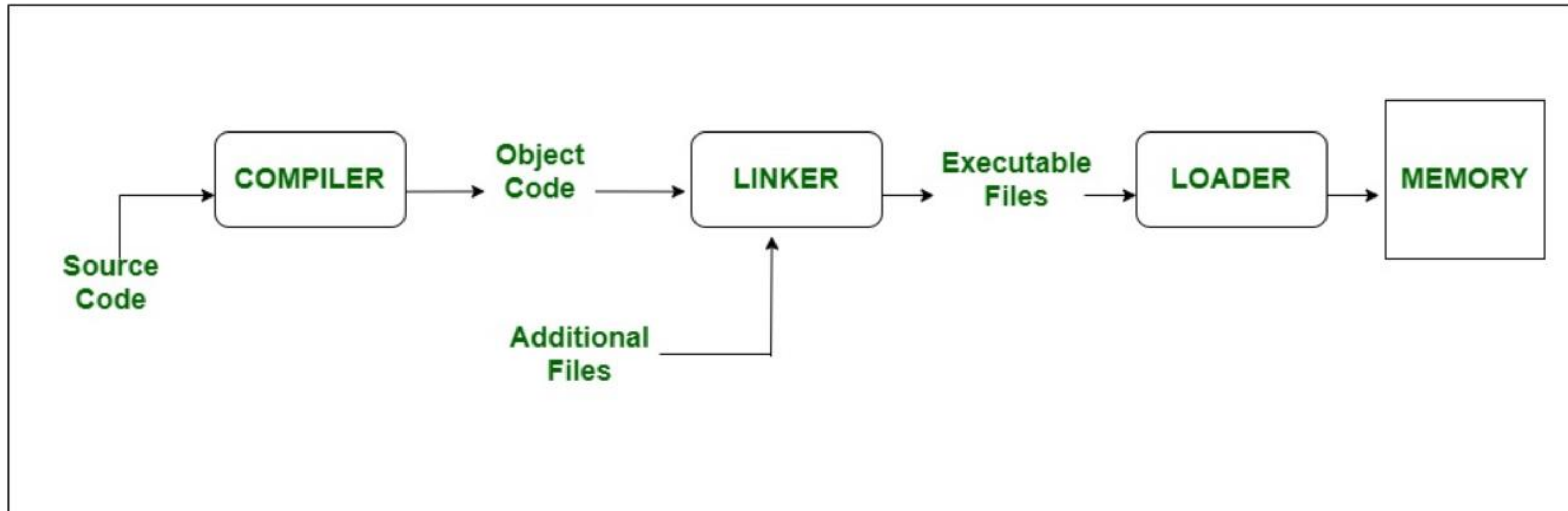
Language

	English	C
Alphabet	A-Z, a-z	A-Z, a-z
Numbers	0-9	0-9, 0 and 1, 0-7, 0-F
Words	Words	Tokens
Sentences	Grammar + Words = Sentences	Syntax + Tokens = Statements
Paragraph	Paragraph	Block
Chapter/Book	Chapter/Book	Program
Library	Library	Library

Compiler and Linker

- **Software Programs**
- **People Language Analogy**
- **Machine (Binary) \Leftrightarrow Operating System (Object Codes) \Leftrightarrow C program (Humans)**
- **Compiler and Linker – Between OS and C Program**
- **Compilation Command**
 - **gcc filename.c**
 - **Creates a.out (Ubuntu) and a.exe (Windows)**
 - **gcc filename.c -o obj**
 - **Creates obj.out (Ubuntu) and obj.exe (Windows)**
- **Compilation fails \Rightarrow Compile Time Error**

Compiler, Linker and Loader



Visual Studio Code

- Demo – First Code

```
void main()  
{  
}
```

Keywords

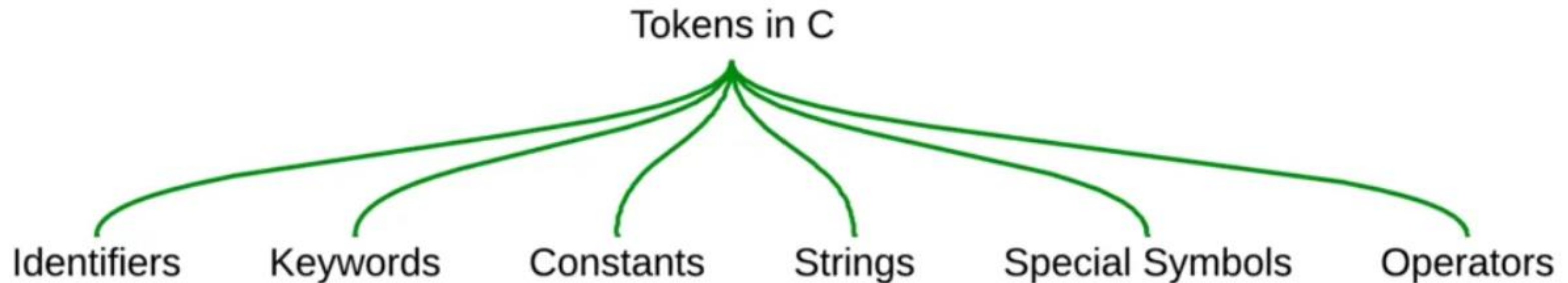
main	void	int	float	char	struct
double	union				

Special Characters

()	{ }			

Types of Tokens

- **Tokens – Smallest unit in a program**
 - **Identifiers**
 - **Keywords**
 - **Constants a.k.a., Literals**
 - **Strings**
 - **Special Symbols or Special Characters**
 - **Operators**



Remember

SHIV NADAR
— UNIVERSITY —
CHENNAI

Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes
- How many numbers can you store?

Keywords

main	void	int	float	char	void
double	union	struct			

Address

10 1000	1000	0000 1010	0000 0000
	1001	0000 0000	0000 0000
	1002	0000 0000	0000 0000
	1003	0000 0000	0000 1010
		Little Endian	Big Endian

Special Characters

()	{}			

Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes
- How many numbers can you store?

Keywords

main	void	int	float	char	void
double	union	struct			

Address

10 1000	1000	0000 1010	0000 0000
	1001	0000 0000	0000 0000
	1002	0000 0000	0000 0000
	1003	0000 0000	0000 1010
		Little Endian	Big Endian

Special Characters

()	{}			

Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes
- How many numbers can you store? - 2^{32}

Address			
<div>10</div> <div>1000</div>	1000	0000 1010	0000 0000
	1001	0000 0000	0000 0000
	1002	0000 0000	0000 0000
	1003	0000 0000	0000 1010
		Little Endian	Big Endian

Keywords

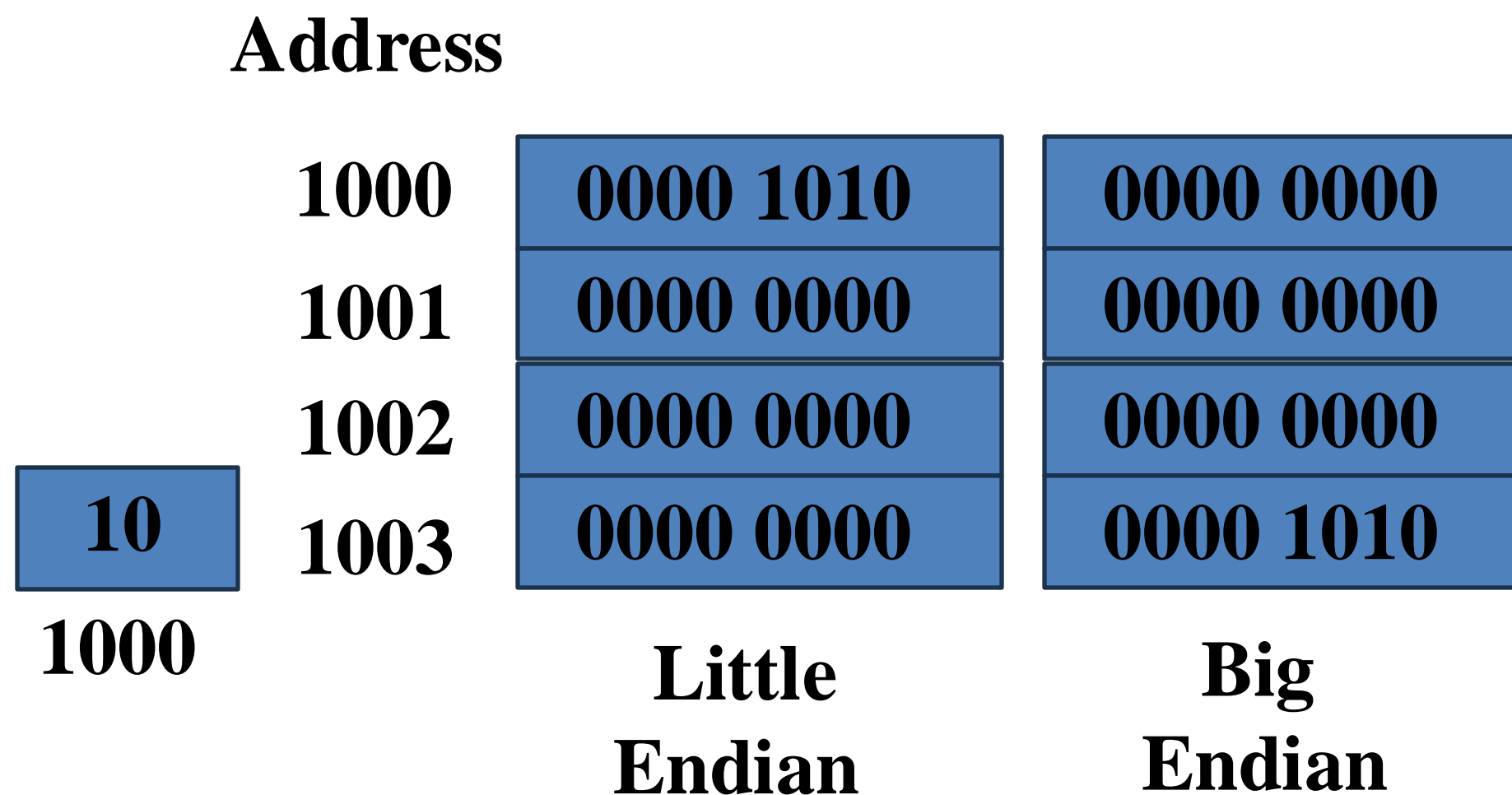
main	void	int	float	char	void
double	union	struct			

Special Characters

()	{ }			

Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes
- How many numbers can you store? - 2^{32}
- 4,294,967,296



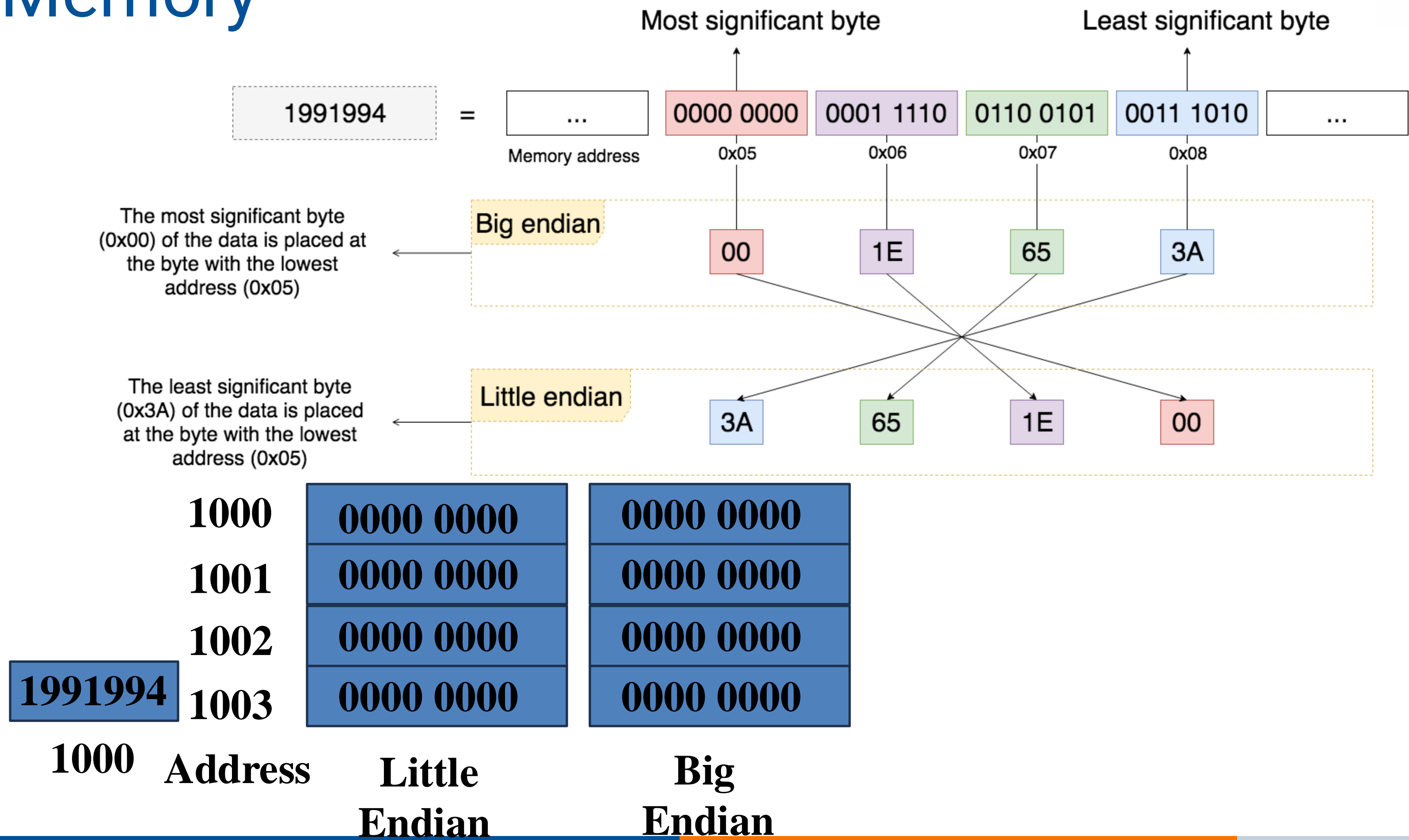
Keywords

main	void	int	float	char	void
double	union	struct			

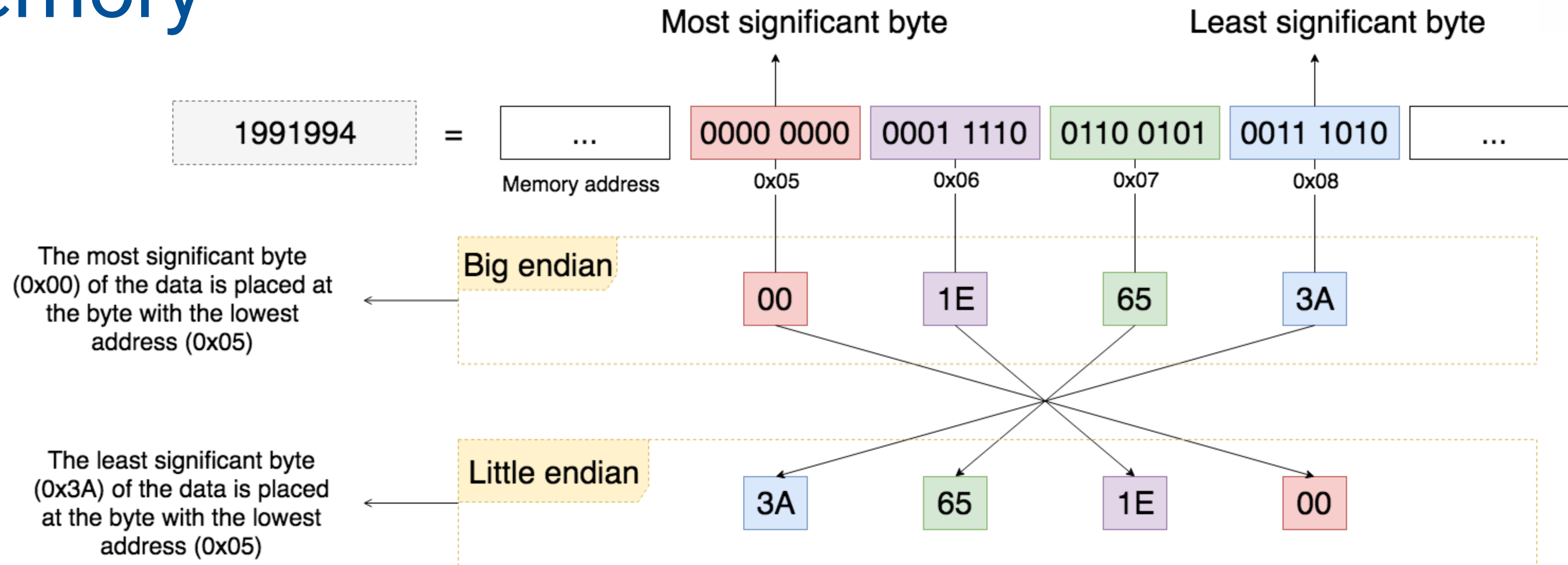
Special Characters

()	{}			

Memory



Memory

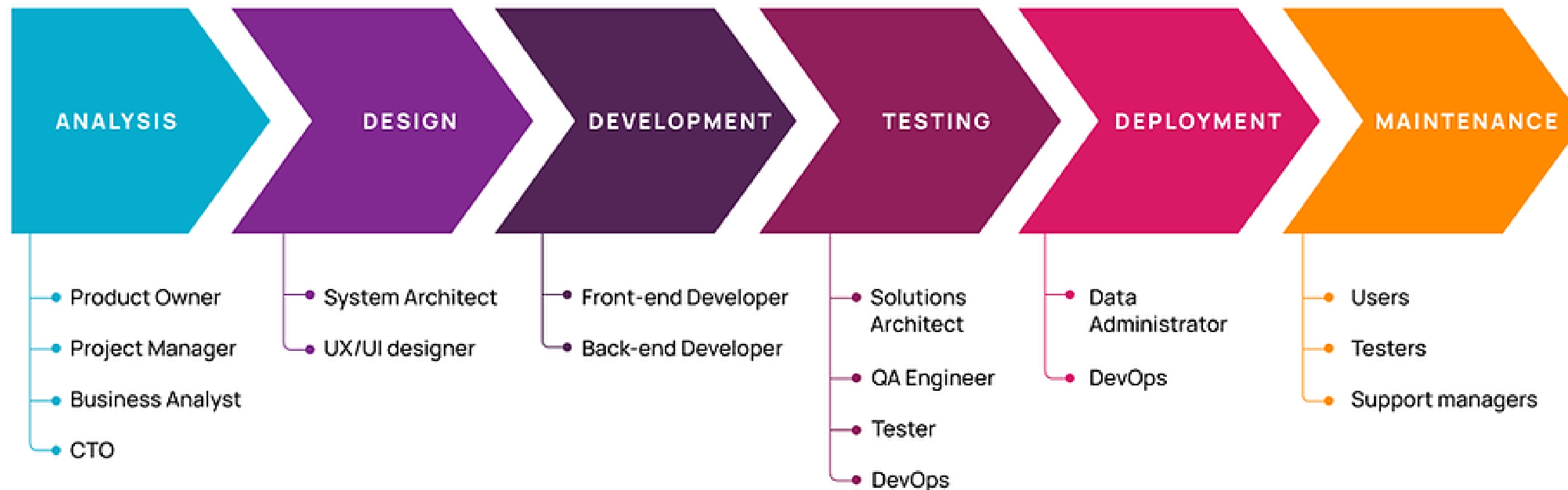


1991994	1000	0000 0000	0011 1010
	1001	0001 1110	0110 0101
	1002	0110 0101	0001 1110
	1003	0011 1010	0000 0000

1000 Address Big Endian Little Endian

Software Development Life-Cycle

6 Phases of the Software Development Life Cycle



Questions?

SHIV NADAR
— UNIVERSITY —
CHENNAI

Algorithm

- **What is it?**

Algorithm

- **What is it?**
 - **Step by step process to solve a problem**
 - **Plan**

Algorithm

- **What is it?**
 - **Step by step process to solve a problem**
 - **Plan**
- **Ways of writing an algorithm**
 - **Informal**
 - **Semi-Formal**
 - **Formal**

Algorithm

- **What is it?**
 - **Step by step process to solve a problem**
 - **Plan**
- **Ways of writing an algorithm**
 - **Informal**
 - **Semi-Formal**
 - **Input**
 - **Output**
 - **Steps**
 - **Formal**

Algorithm

Algorithm 1: Adding two numbers

Input:

Output:

Algorithm

Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output:

Algorithm

Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output: Sum of two numbers

Algorithm

Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output: Sum of two numbers





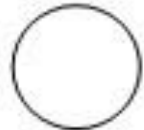

- 1. $\text{sum} = \text{num1} + \text{num2}$**
- 2. return sum**

Flowchart

- **Diagrammatic representation of an algorithm**

Flowchart

- **Diagrammatic representation of an algorithm**

Symbol	Symbol Name
	Terminal
	Process
	Decision
	Input / Output
	Connector
	Flow line

Flowchart

- **Diagrammatic representation of an algorithm**

Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output: Sum of two numbers

1. **$\text{sum} = \text{num1} + \text{num2}$**
2. **return sum**

Flowchart

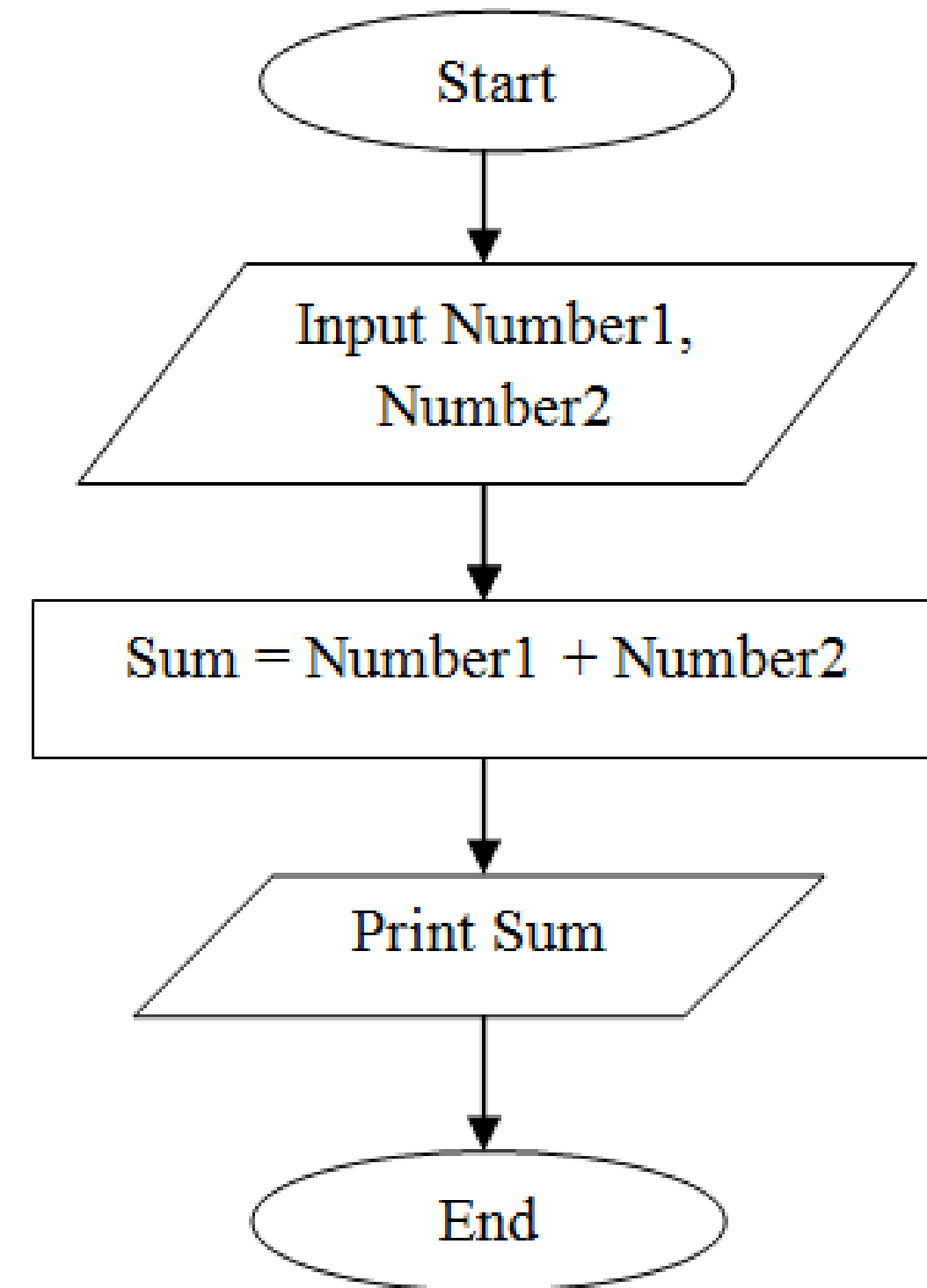
- Diagrammatic representation of an algorithm

Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output: Sum of two numbers

1. $\text{sum} = \text{num1} + \text{num2}$
2. **return sum**



Questions?

SHIV NADAR
— UNIVERSITY —
CHENNAI

Today's Course Outcomes

- **CO1 – Implement C programs from algorithms and flowcharts with error handling. – K3**
- **CO2 – Implement programming fundamentals, decision and looping statements – K3**
- **CO3 – Implement C programs with pointers, arrays, and strings – K3**
- **CO4 – Implement C programs with structures, union, file-handling concepts, and additional features – K3**
- **CO5 – Analyze, breakdown, and solve large computational problems using functions – K4**

Summary

- **Memory**
- **Software Development Lifecycle**
- **Algorithm**
- **Flowchart**
- **Today's Course Outcome**

- **Kernighan, B.W and Ritchie, D. M, “The C Programming language”, 2nd edition, Pearson Education, 2006**

THANK YOU

SHIV NADAR
— UNIVERSITY —
CHENNAI