

# Data Structures



## C Data Types, Memory

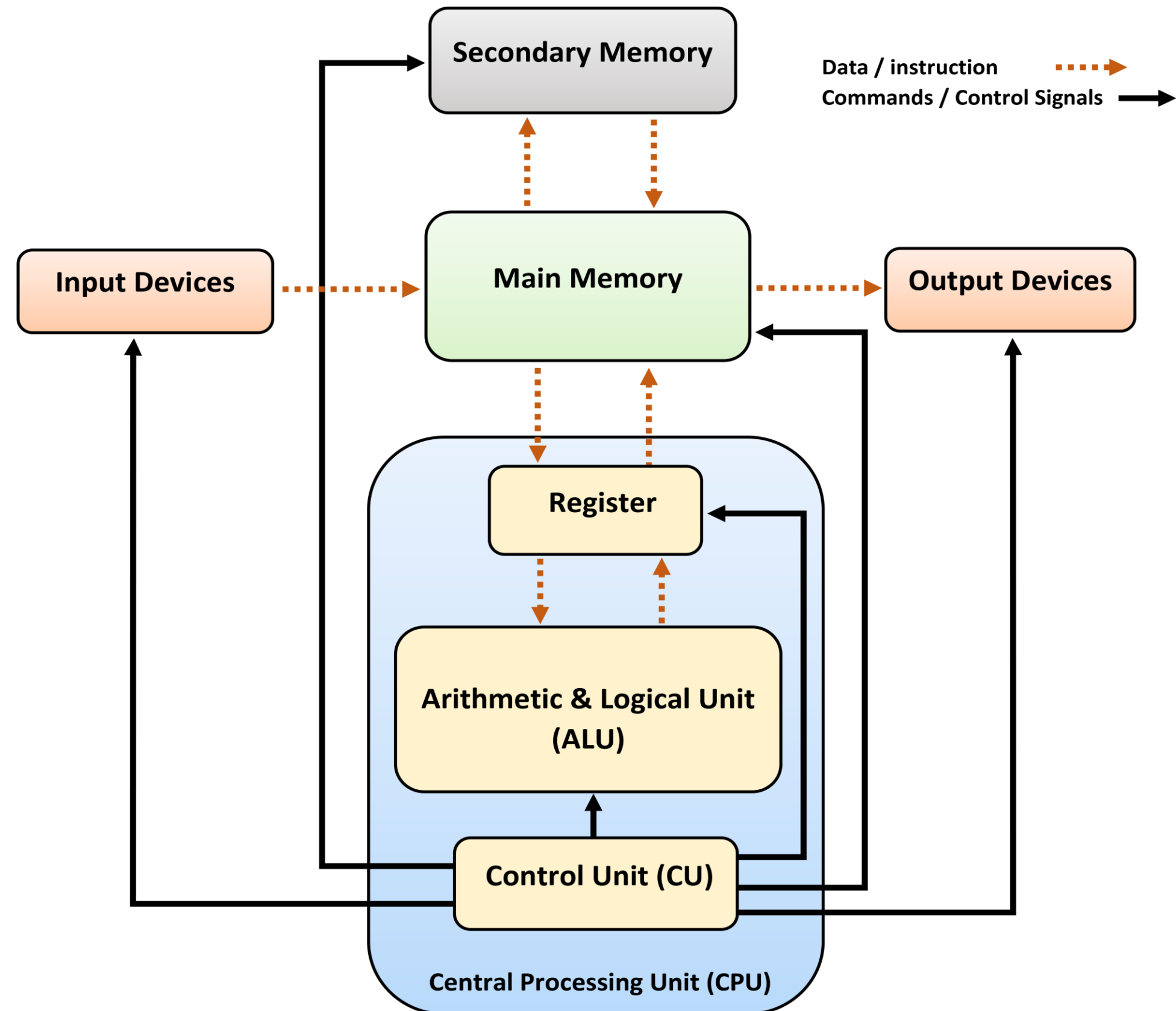
**Subin Sahayam, Assistant Professor,  
Department of Computer Science and Engineering  
Shiv Nadar University**

# Last Class Summary

- **Language**
- **Introduction to Number System**
- **Tokens**
- **Types of Tokens**
- **Visual Studio Code**
- **Compiler and Linker**
- **Compiler, Linker, and Loader**

# General Parts of a Computer

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



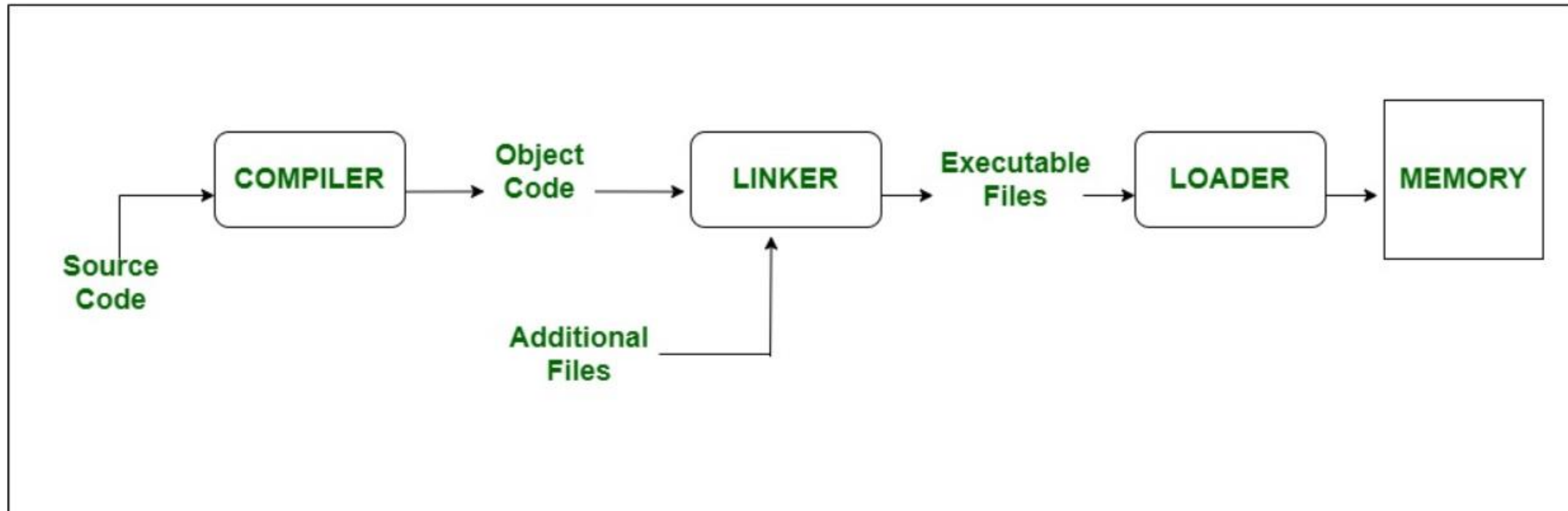
# Language

	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

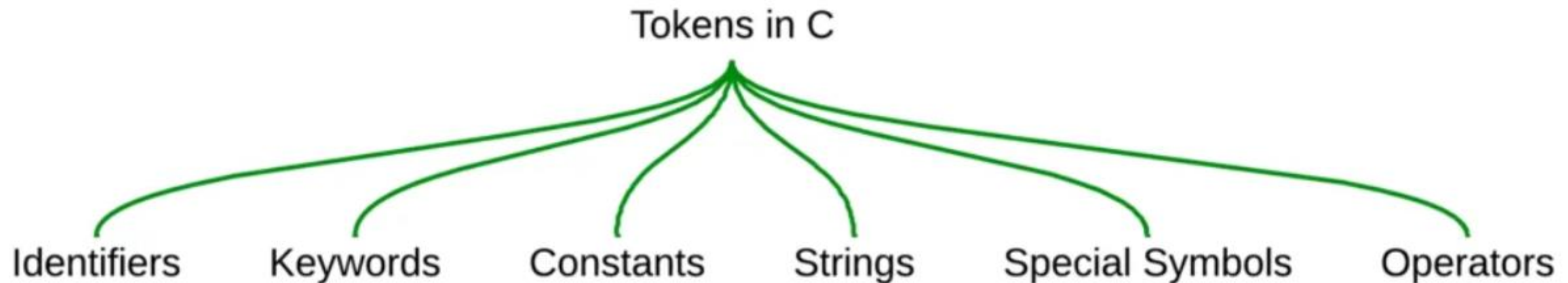
main	void				

## 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**





# Data Types

- **Data?**

## Keywords

main	void				

## Special Characters

()	{ }			

# Data Types

- **Data – Raw Facts**

## Keywords

main	void				

## Special Characters

()	{ }			

# Data Types

- **Data – Raw Facts**
- **Eg: Form data – Name, Age, Height.**

## Keywords

main	void				

## Special Characters

()	{}			

# Data Types

- **Data – Raw Facts**
- **Eg: Form data – Name, Age, Height.**
- **Information?**

## Keywords

main	void				

## Special Characters

()	{ }			

# Data Types

- **Data – Raw Facts**
- **Eg: Form data – Name, Age, Height.**
- **Information – Processed Meaningful Data**

## Keywords

main	void				

## Special Characters

()	{}			

# Data Types

- **Data – Raw Facts**
- **Eg: Form data – Name, Age, Height.**
- **Information – Processed Meaningful Data**
- **Eg: Narendra is 17 years old.**

## Keywords

main	void				

## Special Characters

()	{ }			

# Data Types

- **Data Type – Nature of Raw Facts**

## Keywords

main	void				

## Special Characters

()	{ }			

# Data Types

- **Data Type**
  - **Basic Datatypes**
  - **Derived Datatypes**
  - **Enumeration Datatype**

**Keywords**

main	void				

**Special Characters**

()	{ }			



# Data Types

- **Data Type**
  - **Basic Datatypes**
    - **int**
    - **float**
    - **char**
    - **void**
    - **double**
  - **Derived Datatypes**
  - **Enumeration Datatype**

## Keywords

main	void	int	float	char	void
double					

## Special Characters

()	{ }			

# Data Types

- **Data Type**
  - **Basic Datatypes**
  - **Derived Datatypes**
    - **Arrays**
    - **Functions**
    - **Pointers**
    - **Unions**
    - **Structures**
  - **Enumeration Datatype**

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			

# Data Types

- **Data Type**
  - **Basic Datatypes**
  - **Derived Datatypes**
  - **Enumeration Datatype**
    - **enum**

Keywords

main	void	int	float	char	void
double	union	struct	enum		

Special Characters

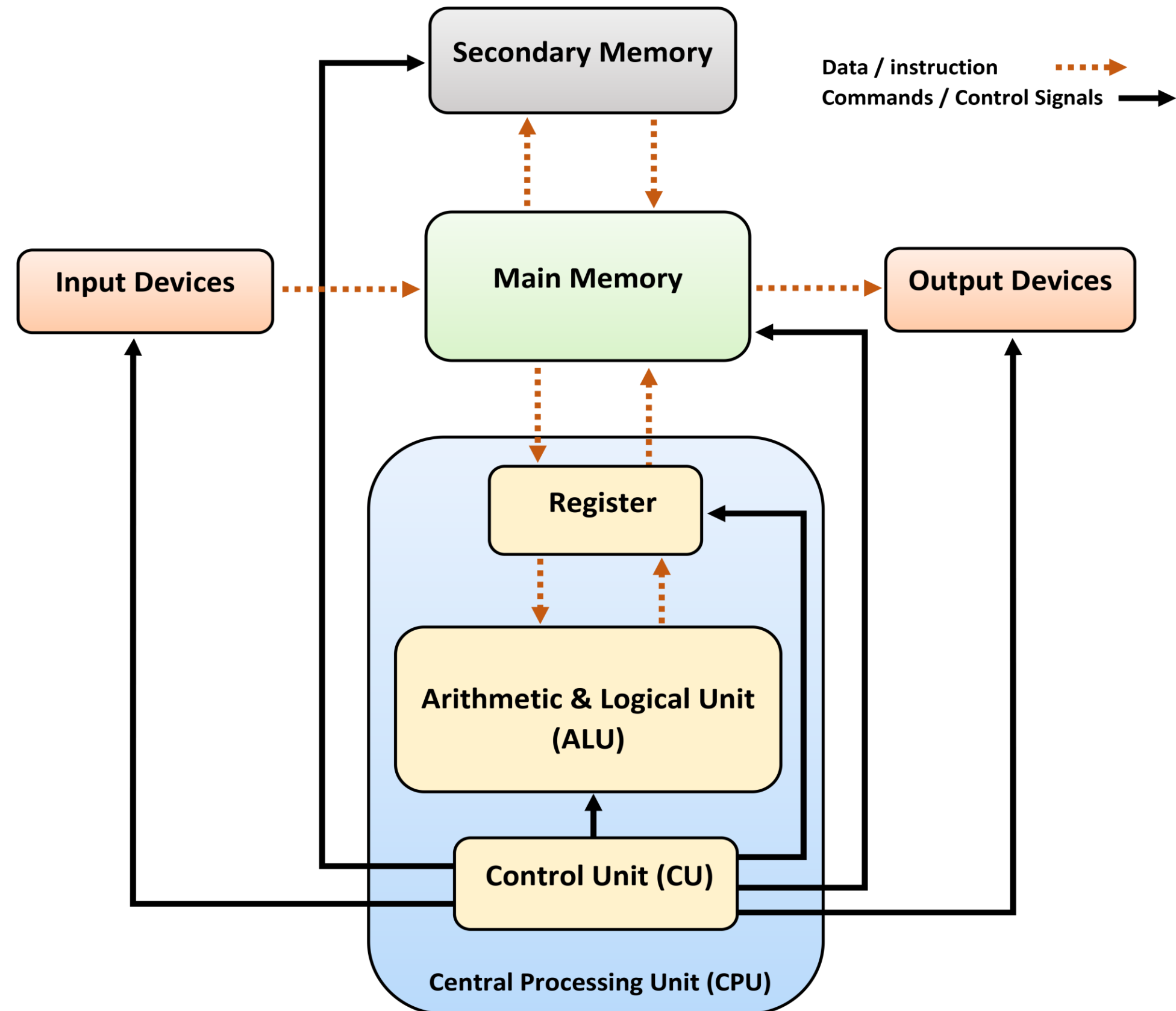
()	{ }			

# Questions?

**SHIV NADAR**  
— UNIVERSITY —  
CHENNAI

# General Parts of a Computer

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



# Memory

- Adding two numbers.
- What do you need?

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			

# Memory

- Adding two numbers.
- What do you need?
  - Two Numbers

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			

# Memory

- Adding two numbers.
- What do you need?
  - Two Numbers
  - Storage

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			



# Memory

- Adding two numbers.
- What do you need?
  - Two Numbers
  - Storage
    - RAM
      - Memory Location

10

1000

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			

# Memory

- Adding two numbers.
- What do you need?
  - Two Numbers
  - Storage
    - RAM
      - Memory Location
      - Binary of 10?

10

1000

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			

# Memory

- Adding two numbers.
- What do you need?
  - Two Numbers
  - Storage
    - RAM
      - Memory Location
      - Binary of 10 -  $(1010)_2$

10

1000

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			

# Memory

- Adding two numbers.
- What do you need?
  - Two Numbers
  - Storage
    - RAM
      - Memory Location
      - Binary of 10 -  $(1010)_2$
      - **Binary Digit** - Bit

10

1000

## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			

# Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte

0000 0000

10

1000

## Keywords

main	void	int	float	char	void
double	union	struct			

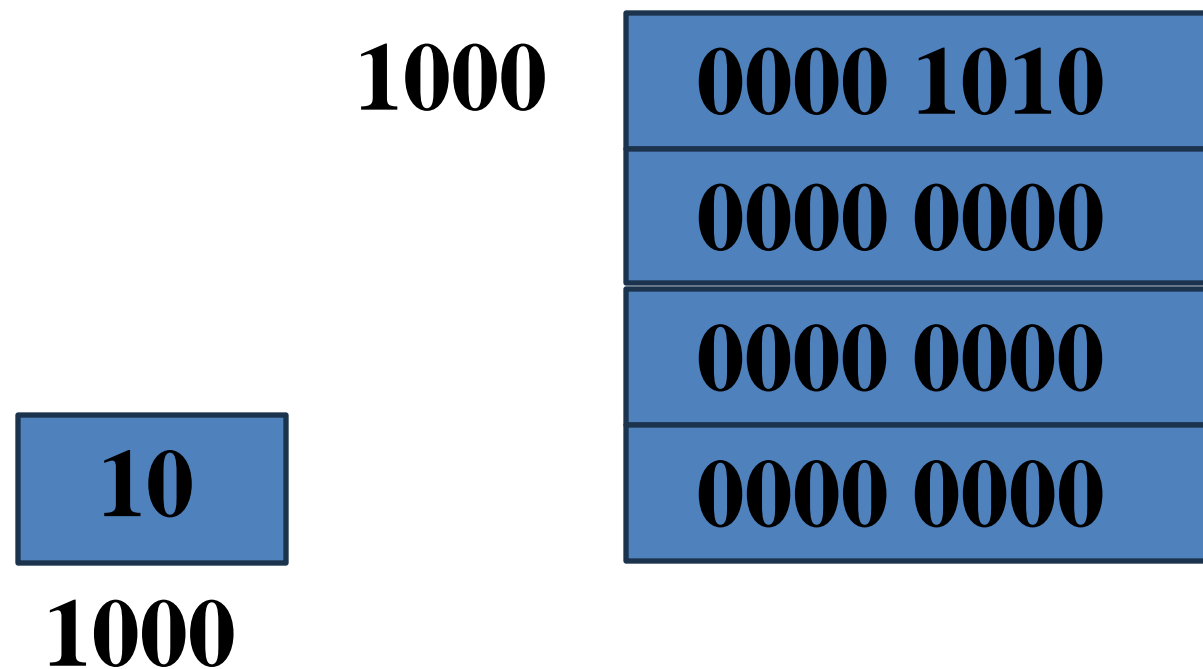
## Special Characters

()	{ }			

# Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes

## Address



## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{}			

# Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes

## Address

<div>10</div> 1000	1000	0000 1010
	1001	0000 0000
	1002	0000 0000
	1003	0000 0000

## Keywords

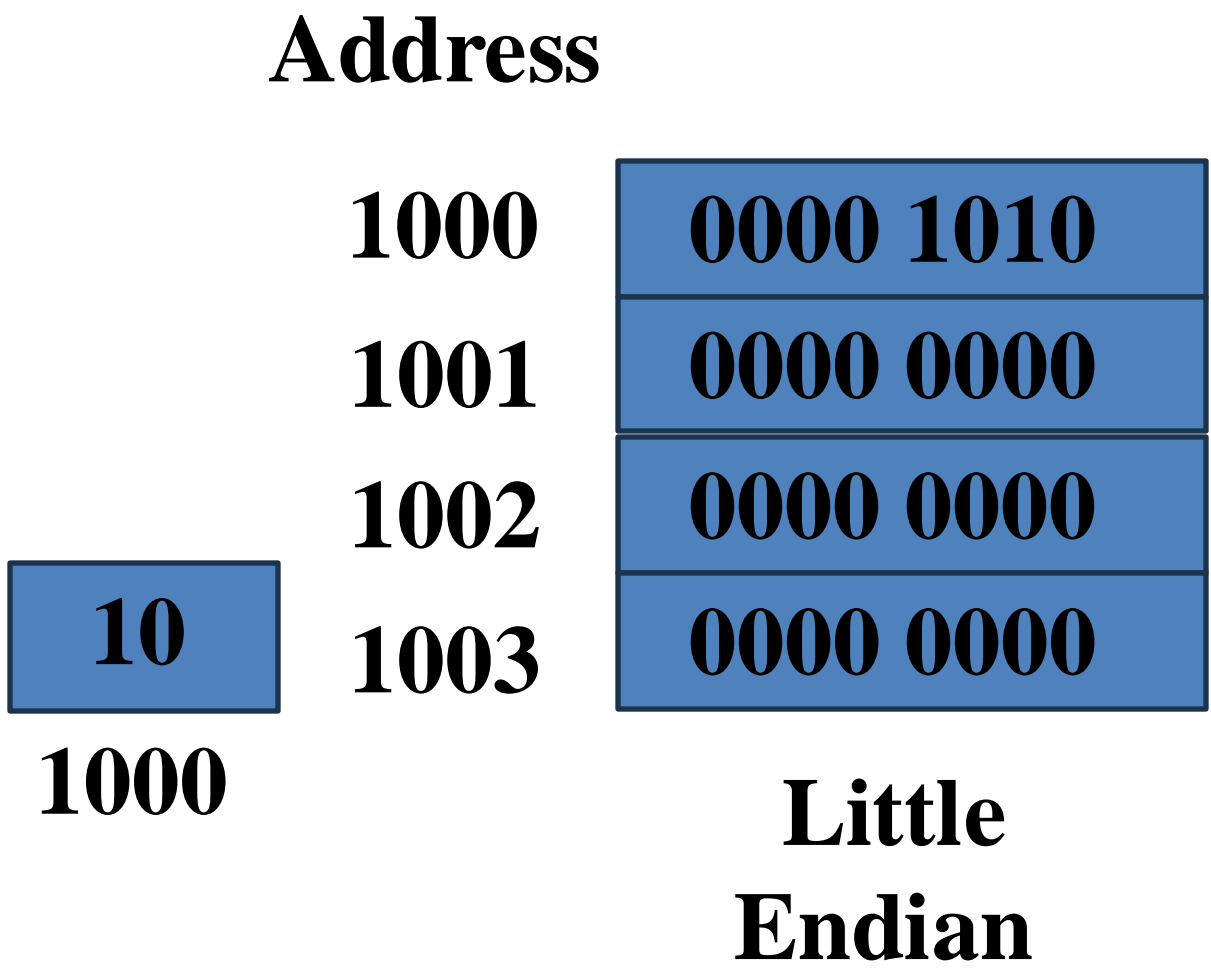
main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{}			

# Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes



## Keywords

main	void	int	float	char	void
double	union	struct			

## Special Characters

()	{ }			



# Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes

## Keywords

main	void	int	float	char	void
double	union	struct			

## Address

10	1000	0000 1010	0000 0000
	1001	0000 0000	0000 0000
	1002	0000 0000	0000 0000
	1003	0000 0000	0000 1010

1000

Little  
Endian

## Special Characters

()	{}			

# Memory

- **Binary Digit** – Bit
- 8 Bits – 1 Byte
- Assume numbers => 4 Bytes

## Keywords

main	void	int	float	char	void
double	union	struct			

## 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

## Special Characters

()	{}			

# 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

- **Data types**
- **Memory**
- **Today's Course Outcome**

# References

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

THANK YOU

SHIV NADAR  
— UNIVERSITY —  
CHENNAI