Data Structures

Variable Declaration

Subin Sahayam, Assistant Professor,

Department of Computer Science and Engineering
Shiv Nadar University

Last Class Summary

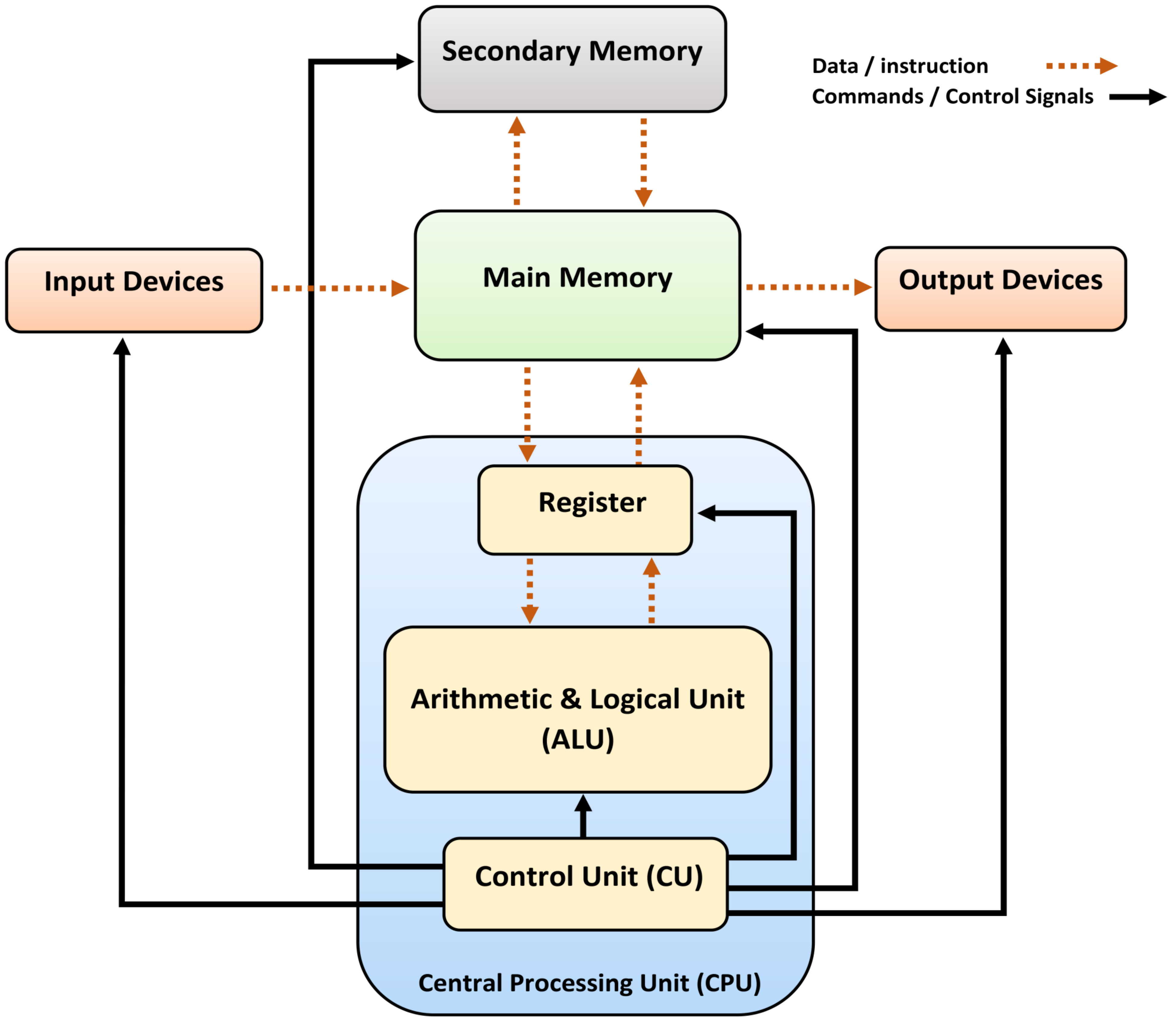
- **Memory**
- **Software Development Lifecycle**
- **Algorithm**
- **Flowchart**

Remember

SHIV NADAR UNIVERSITY CHENNAI

General Parts of a Computer

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



K1

Language

SHIV NADAR UNIVERSITY CHENNAI

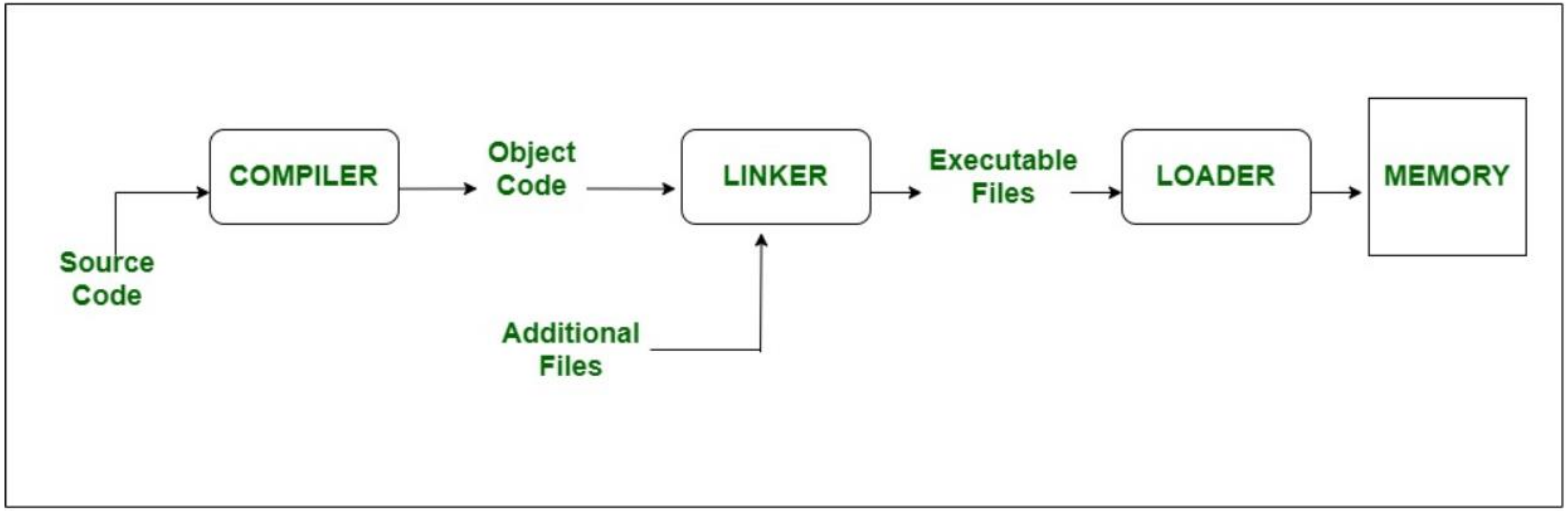
	English	C
Alphabet	A-Z, a-z	A-Z, a-z
Numbers	0-9	0-9, 0 and 1, 0-7, 0-F
Words	Words	Tokens
Sentences	Grammar + Words = Sentences	Syntax + Tokens = Statements
Paragraph	Paragraph	Block
Chapter/Book	Chapter/Book	Program
Library	Library	Library

K1

Compiler and Linker

- **Software Programs**
- **People Language Analogy**
- **Machine (Binary) <=> Operating System (Object Codes) <=> 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 => Compile Time Error**

Compiler, Linker and Loader



K1

Visual Studio Code

- **Demo – First Code**

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

Keywords

main	void	int	float	char	struct
double	union				

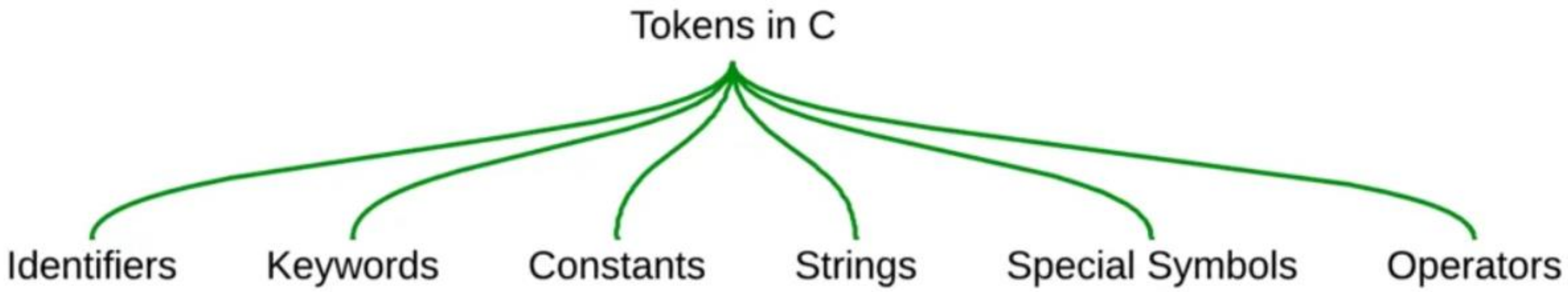
Special Characters

()	{}			

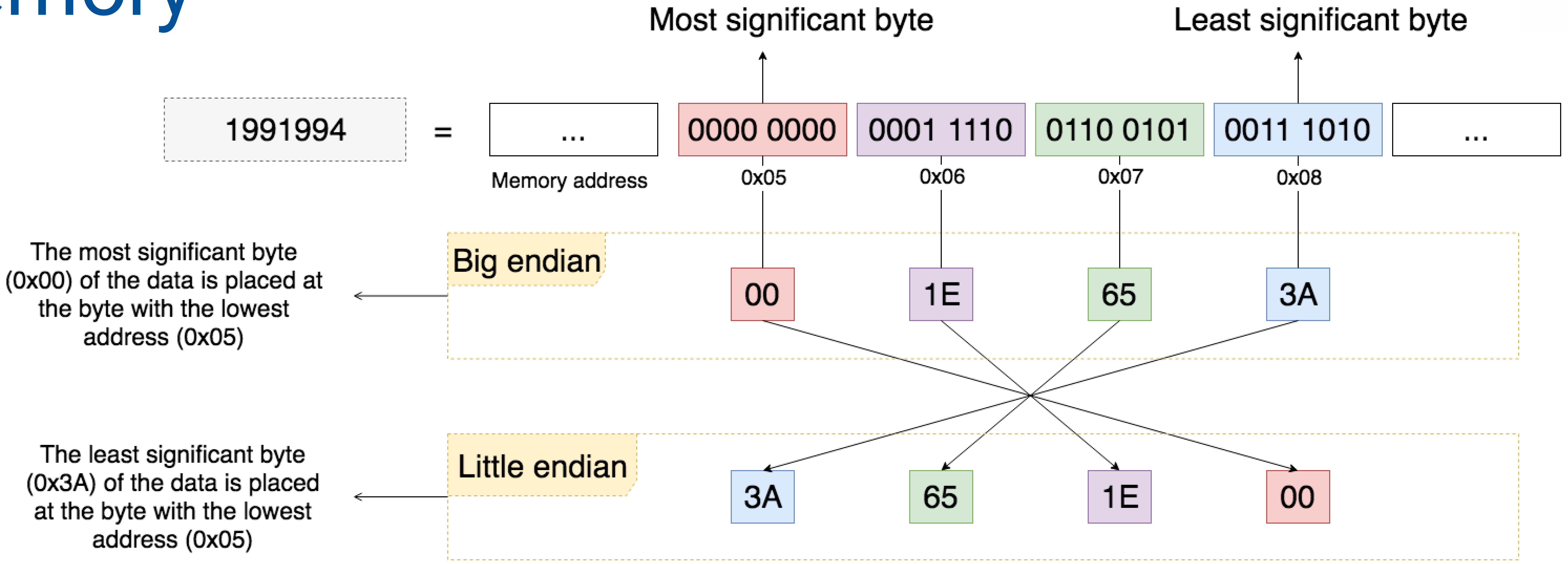
K1

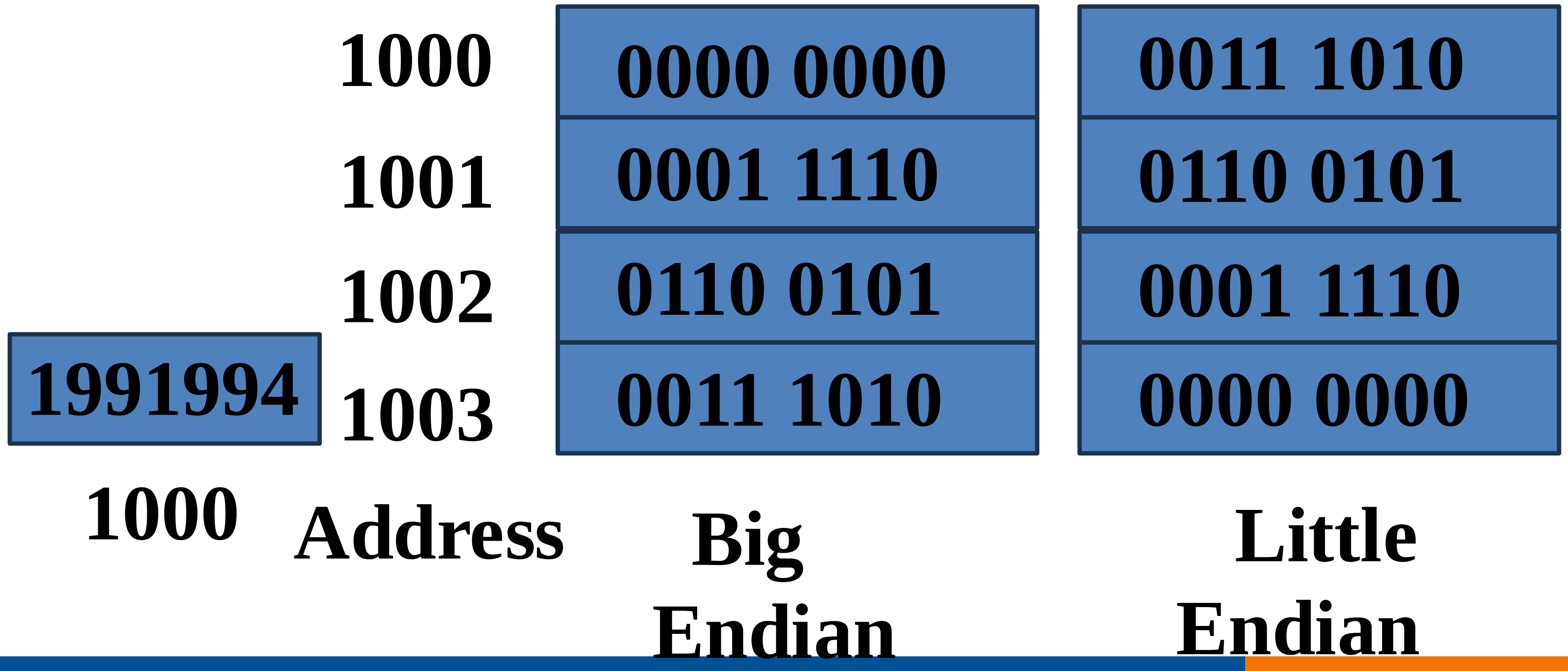
Types of Tokens

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



Memory


Most significant byte
Least significant byte
1991994
=
...
0000 0000
0001 1110
0110 0101
0011 1010
...
Memory address
0x05
0x06
0x07
0x08
The most significant byte (0x00) of the data is placed at the byte with the lowest address (0x05)
Big endian
00
1E
65
3A
The least significant byte (0x3A) of the data is placed at the byte with the lowest address (0x05)
Little endian
3A
65
1E
00

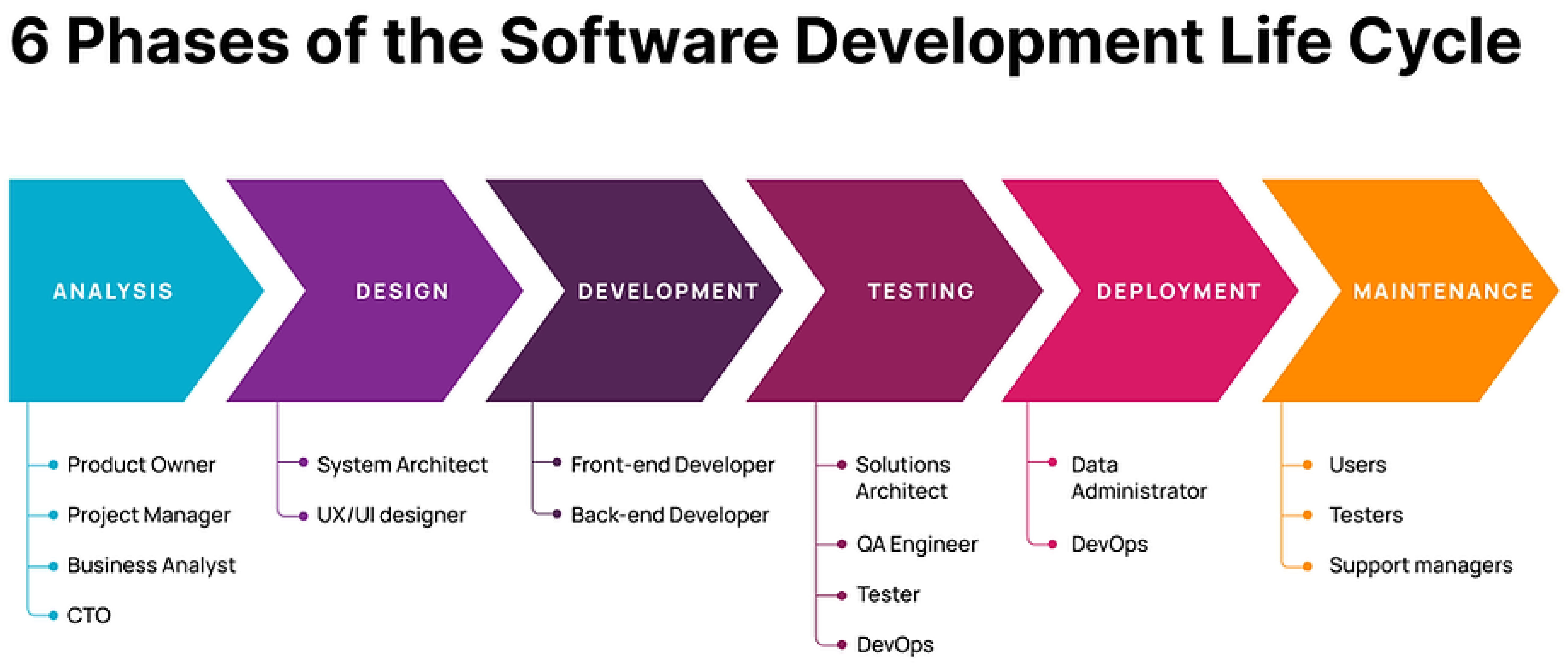

1991994
1000

Address	Big Endian	Little Endian
1000	0000 0000	0011 1010
1001	0001 1110	0110 0101
1002	0110 0101	0001 1110
1003	0011 1010	0000 0000

K2

Software Development Life-Cycle

SHIV NADAR UNIVERSITY CHENNAI



K2

Remember

SHIV NADAR UNIVERSITY CHENNAI

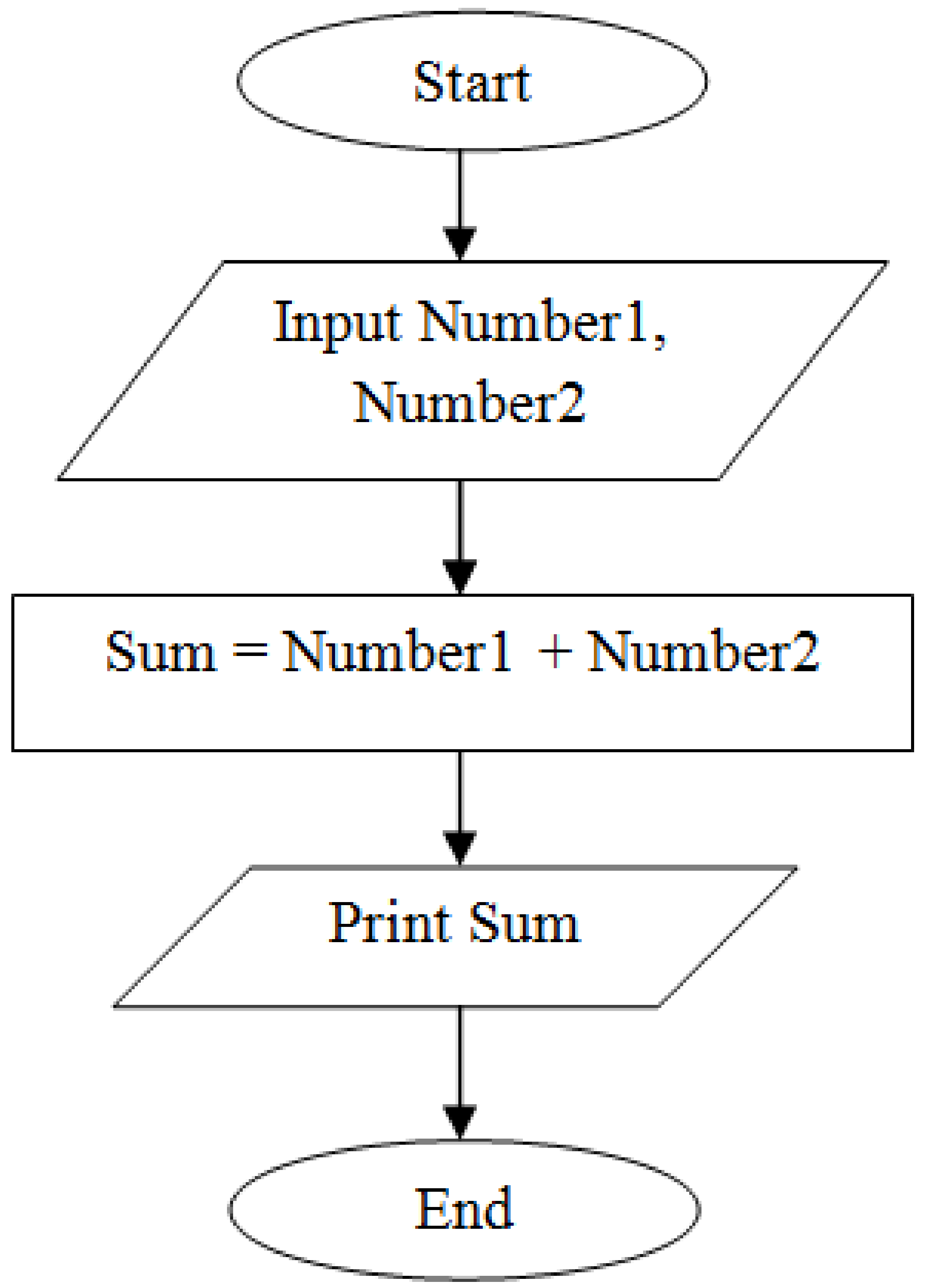
Program

Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.  sum = num1 + num2
2.  return sum
```



Program – Variable Declaration

- **Who does process in your computer?**

K2

Program – Variable Declaration

- **Who does process in your computer?**
  - **CPU**

K2

Program – Variable Declaration

- **Who does process in your computer?**
  - **CPU**
- **How do you access the CPU?**

K2

Program – Variable Declaration

- **Who does process in your computer?**
  - **CPU**
- **How do you access the CPU?**
  - **Main Memory or RAM**

K2

Program – Variable Declaration

- **Who does process in your computer?**
  - **CPU**
- **How do you access the CPU?**
  - **Main Memory or RAM**
- **Who manages the RAM?**

K2

Program – Variable Declaration

- **Who does process in your computer?**
  - **CPU**
- **How do you access the CPU?**
  - **Main Memory or RAM**
- **Who manages the RAM?**
  - **Operating System (OS)**

K2

Program – Variable Declaration

- **Who does process in your computer?**
  - **CPU**
- **How do you access the CPU?**
  - **Main Memory or RAM**
- **Who manages the RAM?**
  - **Operating System (OS)**
- **So, to give input the computer to perform any operation or process, what should you do?**

K2

Program – Variable Declaration

- **Who does process in your computer?**
    - **CPU**
- **How do you access the CPU?**
    - **Main Memory or RAM**
- **Who manages the RAM?**
    - **Operating System (OS)**
- **So, to give input the computer to perform any operation or process, what should you do?**
    - **Request memory location from OS**

K2

Program – Variable Declaration

- **Who does process in your computer?**
  - **CPU**
- **How do you access the CPU?**
  - **Main Memory or RAM**
- **Who manages the RAM?**
  - **Operating System (OS)**
- **So, to give input the computer to perform any operation or process, what should you do?**
  - **Request memory location from OS**
  - **Done with the help of variable declaration**

K2

Language

SHIV NADAR UNIVERSITY CHENNAI

	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

K1

Program – Variable Declaration

- **Variables are name of locations**
- **Store data**
- **Declaration Statement**

Keywords

main	**void**	**int**	**float**	**char**	**struct**
double	**union**				

Special Characters

()	**{}**			

K1

Program – Variable Declaration

- **Declaration Statement**
- **Syntax**
  - **datatype identifier1, .. , identifier n;**

Keywords

main	**void**	**int**	**float**	**char**	**struct**
double	**union**				

Special Characters

()	**{}**			

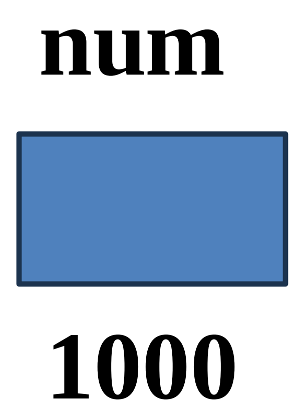
K1

Program – Variable Declaration

- Declaration Statement
- Syntax
  - datatype identifier1, .. , identifier n;
- Eg:
  - int num;

Keywords

main	void	int	float	char	struct
double	union				



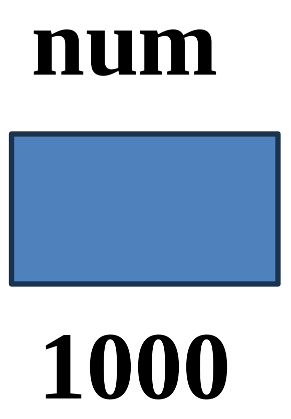
Special Characters

()	{}			

K2

Program – Variable Declaration

- Declaration Statement
- Syntax
  - datatype identifier1, .. , identifier n;
- Eg:
  - int num;
- int – 4 bytes
- float – 4 bytes
- char – 1 byte
- double – 8 bytes



Keywords

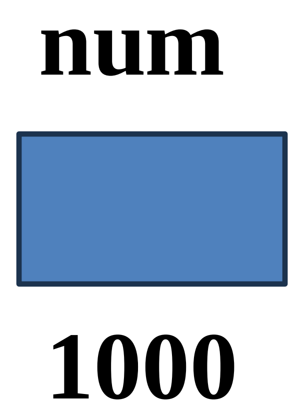
main	void	int	float	char	struct
double	union				

Special Characters

()	{}			

Program – Variable Declaration

- Declaration Statement
- Syntax
  - datatype identifier1, .. , identifier n;
- Eg:
  - int num;
- = Assignment Operator



Keywords

main	void	int	float	char	struct
double	union				

Special Characters

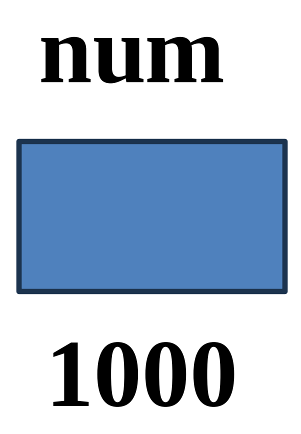
()	{}			

K1

Program – Variable Declaration

SHIV NADAR UNIVERSITY CHENNAI

- **Declaration Statement**
- **Syntax**
  - **datatype identifier1, .. , identifier n;**
- **Eg:**
  - **int num;**
- **= Assignment Operator**
  - **Assignment Statement**
  - **Initialization Statement**



Keywords

main	void	int	float	char	struct
double	union				

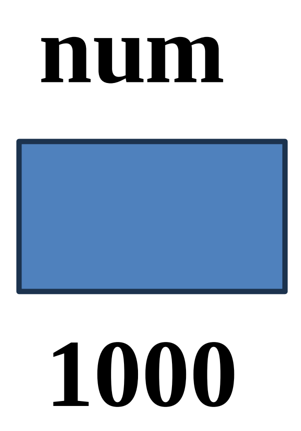
Special Characters

()	{}			

K1

Program – Variable Declaration

- **Declaration Statement**
- **Syntax**
  - **datatype identifier1, .. , identifier n;**
- **Eg:**
  - **int num;**
- **= Assignment Operator**
  - **Assignment Statement**
  - **Initialization Statement**
    - **First time assignment**



Keywords

main	void	int	float	char	struct
double	union				

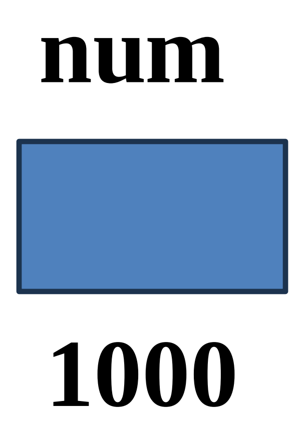
Special Characters

()	{}			

K1

Program – Variable Declaration

- **Declaration Statement**
- **Syntax**
  - **datatype identifier1, .. , identifier n;**
- **Eg:**
  - **int num;**
- **= Assignment Operator**
  - **Assignment Statement**
  - **Initialization Statement**
- **Syntax**
  - **identifier = constants/literals;**
  - **identifier = identifier;**



Keywords

main	void	int	float	char	struct
double	union				

Special Characters

()	{}			

K1

Program – Variable Declaration

- **= Assignment Operator**
- **Syntax**
  - **identifier = constants/literals;**
  - **identifier1 = identifier2;**
- **Eg:**
  - **int num1, num2 = 15;**

Keywords

main	void	int	float	char	struct
double	union				

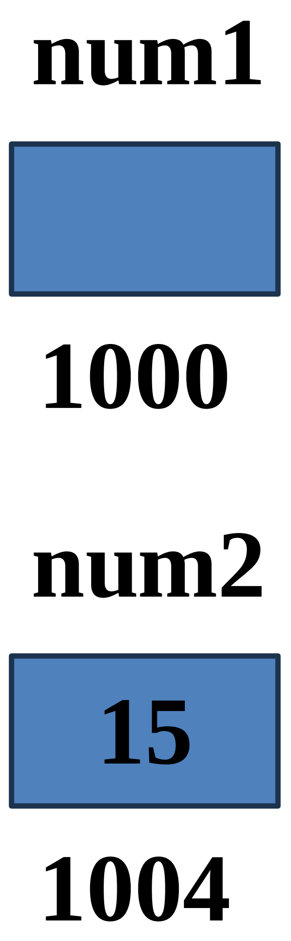
Special Characters

()	{}			

K2

Program – Variable Declaration

- = Assignment Operator
- Syntax
  - identifier = constants/literals;
  - identifier1 = identifier2;
- Eg:
  - int num1, num2 = 15;



Keywords

main	void	int	float	char	struct
double	union				

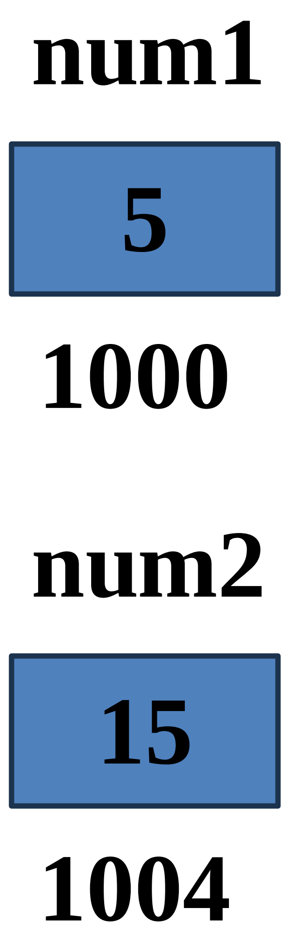
Special Characters

()	{}			

K2

Program – Variable Declaration

- = Assignment Operator
- Syntax
  - identifier = constants/literals;
  - identifier1 = identifier2;
- Eg:
  - int num1, num2 = 15;
  - num1 = 5; //Initialization



Keywords

main	void	int	float	char	struct
double	union				

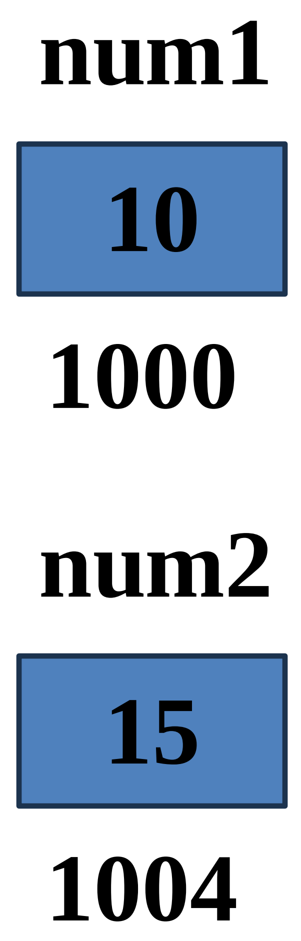
Special Characters

()	{}			

K2

Program – Variable Declaration

- = Assignment Operator
- Syntax
  - identifier = constants/literals;
  - identifier1 = identifier2;
- Eg:
  - int num1, num2 = 15;
  - num1 = 5; //Initialization
  - num1 = 10; // Assignment



Keywords

main	void	int	float	char	struct
double	union				

Special Characters

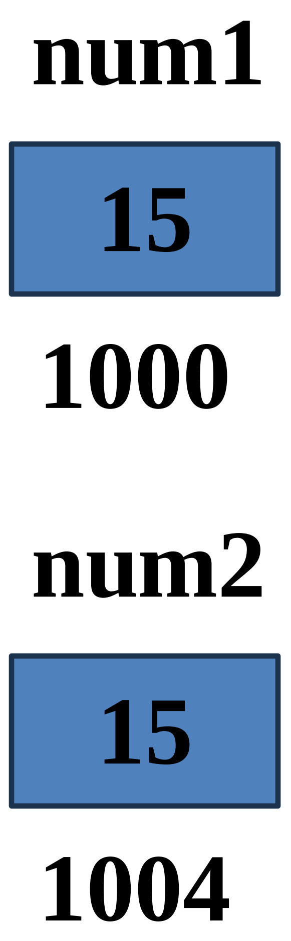
()	{}			

K2

SHIV NADAR UNIVERSITY CHENNAI

Program – Variable Declaration

- **= Assignment Operator**
- **Syntax**
  - **identifier = constants/literals;**
  - **identifier1 = identifier2;**
- **Eg:**
  - **int num1, num2 = 15;**
  - **num1 = 5; //Initialization**
  - **num1 = 10; // Assignment**
  - **num1 = num2; // Assignment**



Keywords

main	void	int	float	char	struct
double	union				

Special Characters

()	{}			

K2

Program

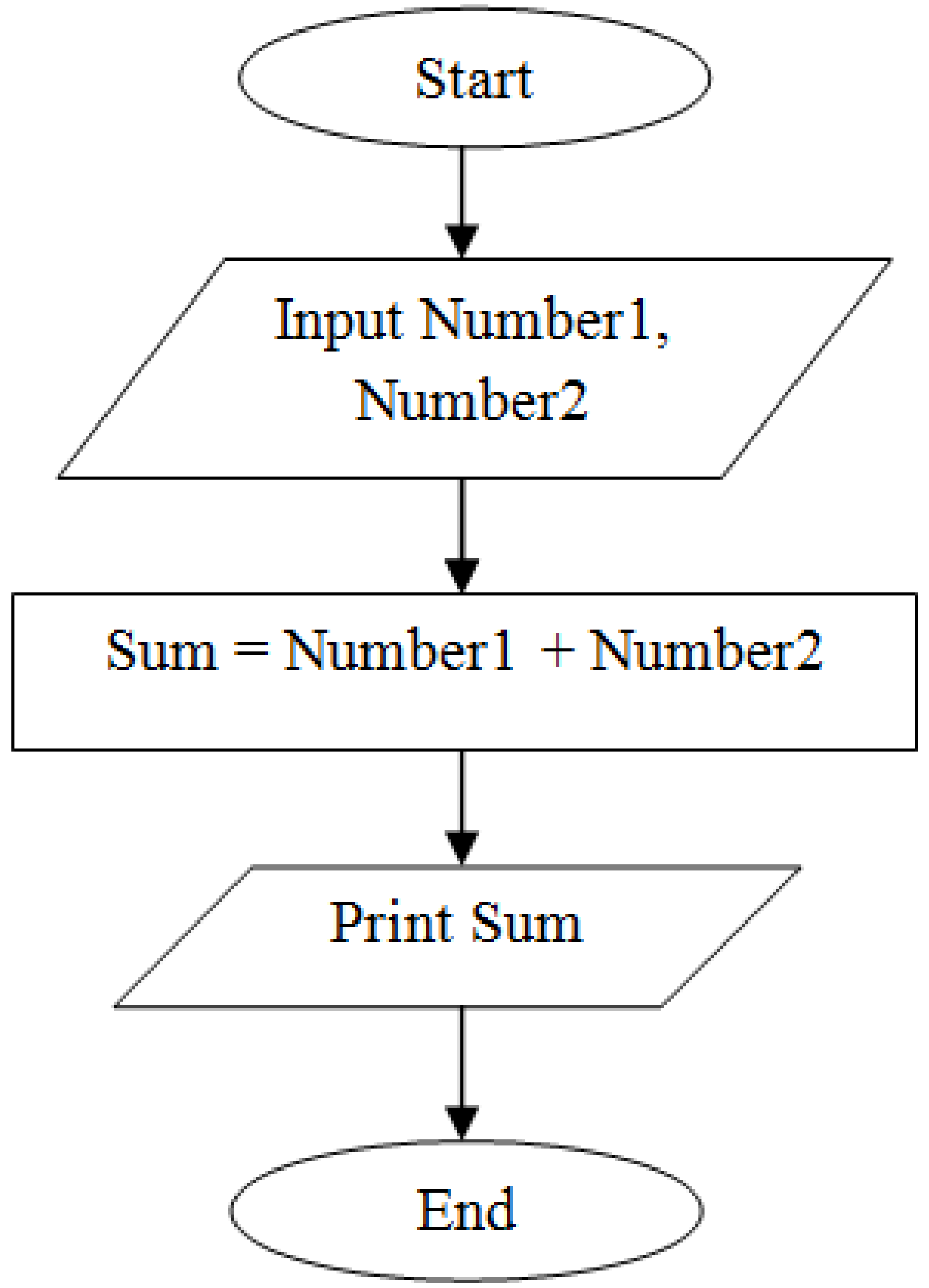
Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.  sum = num1 + num2
2.  return sum
```

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



K2

Program

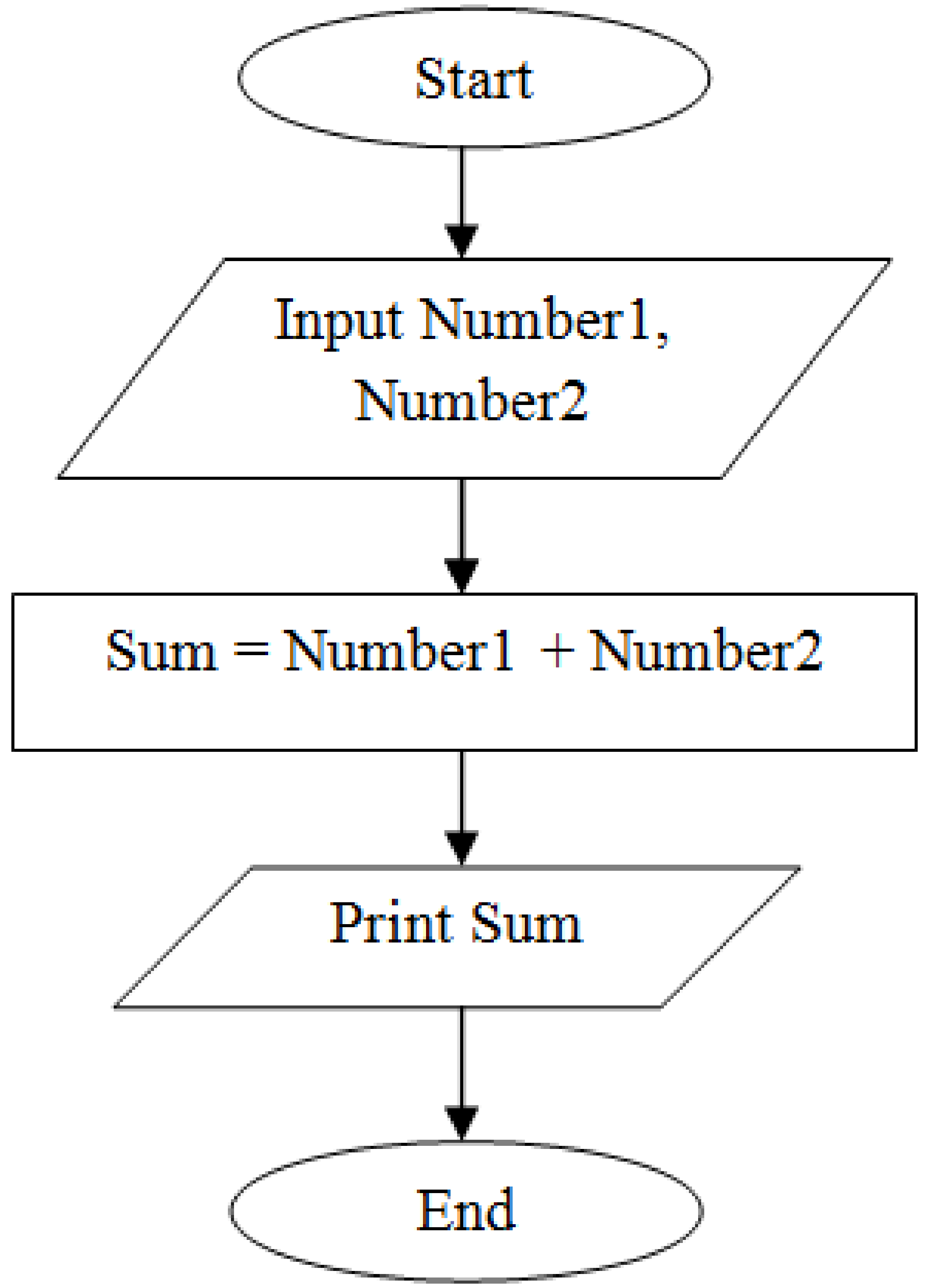
Algorithm 1: Adding two numbers

Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.  sum = num1 + num2
2.  return sum
```

```
void main()
{
   int num1, num2, sum;
}
```



K3

Program

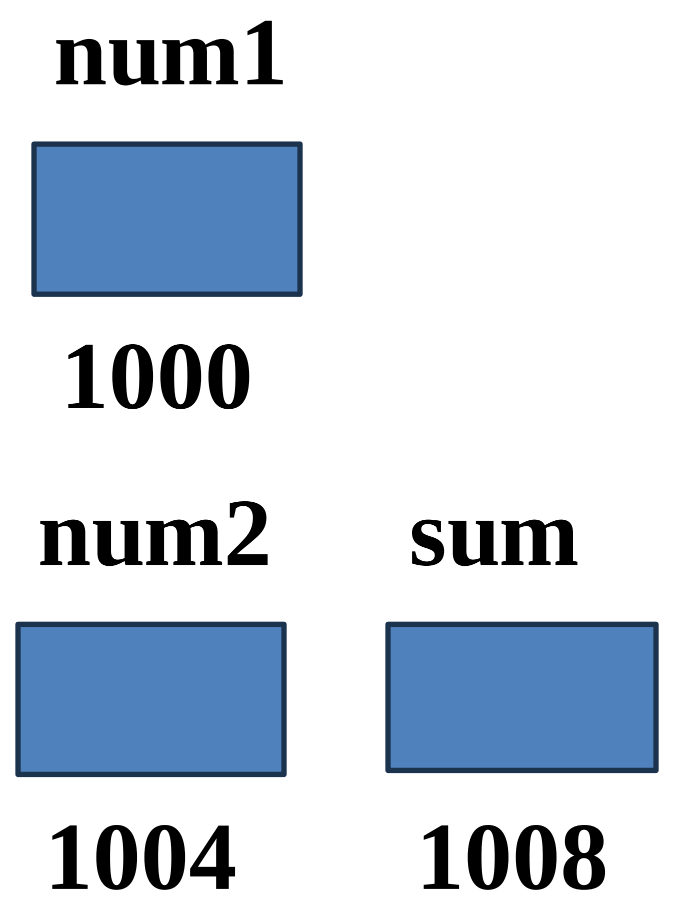
Algorithm 1: Adding two numbers

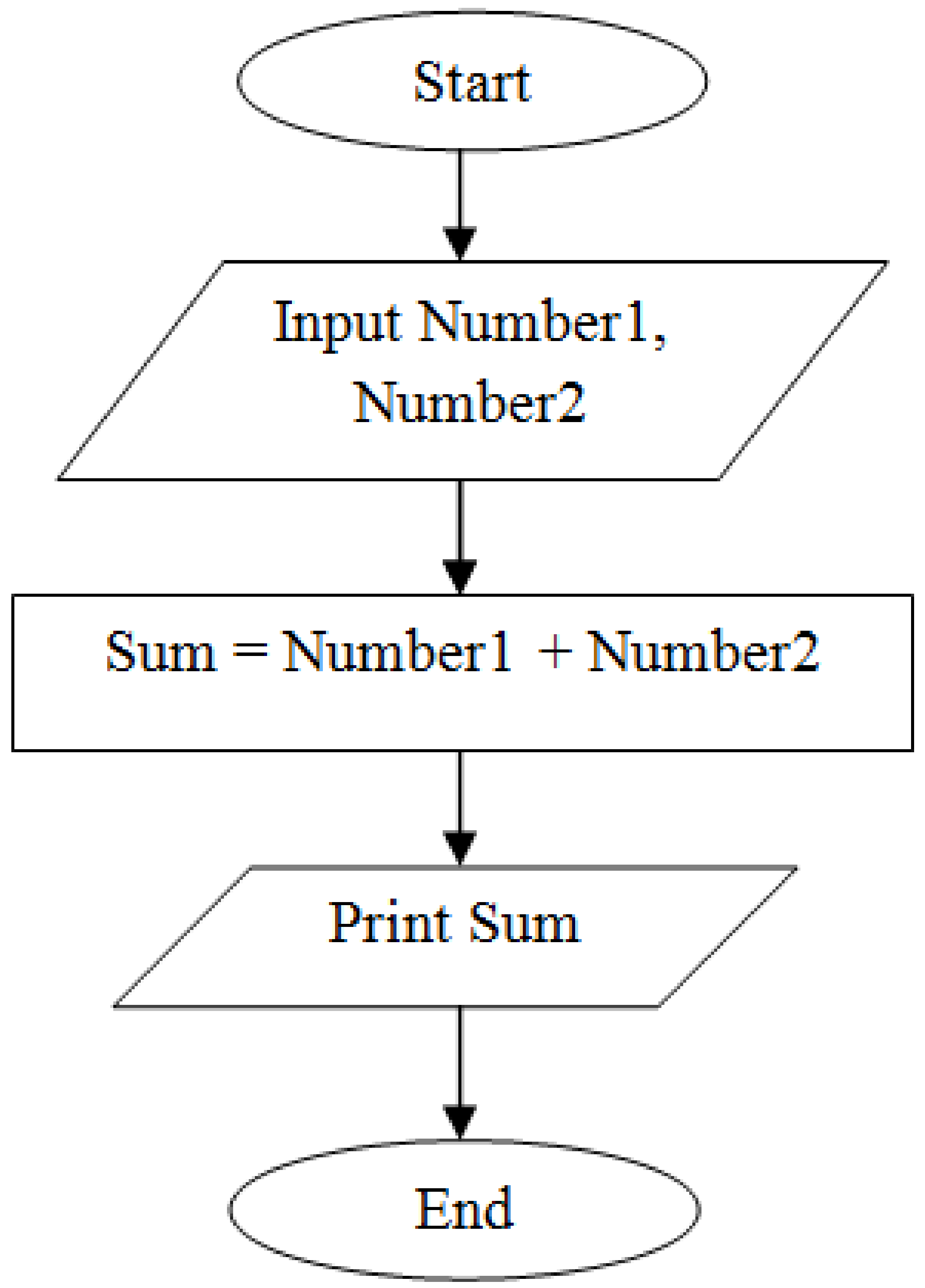
Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.   sum = num1 + num2
2.   return sum
```

```
void main()
{
   int num1, num2, sum;
}
```





K3

Program

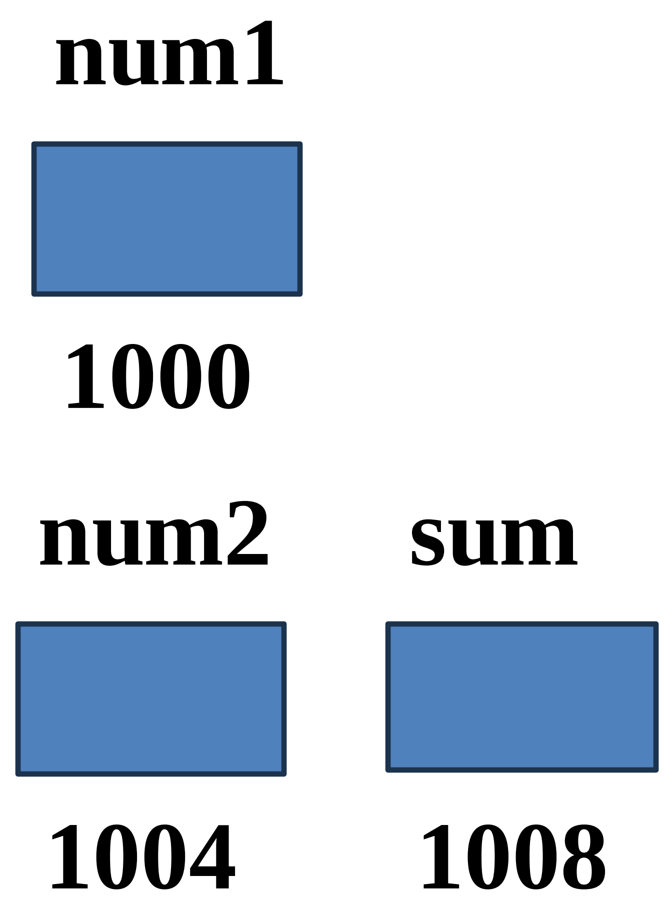
Algorithm 1: Adding two numbers

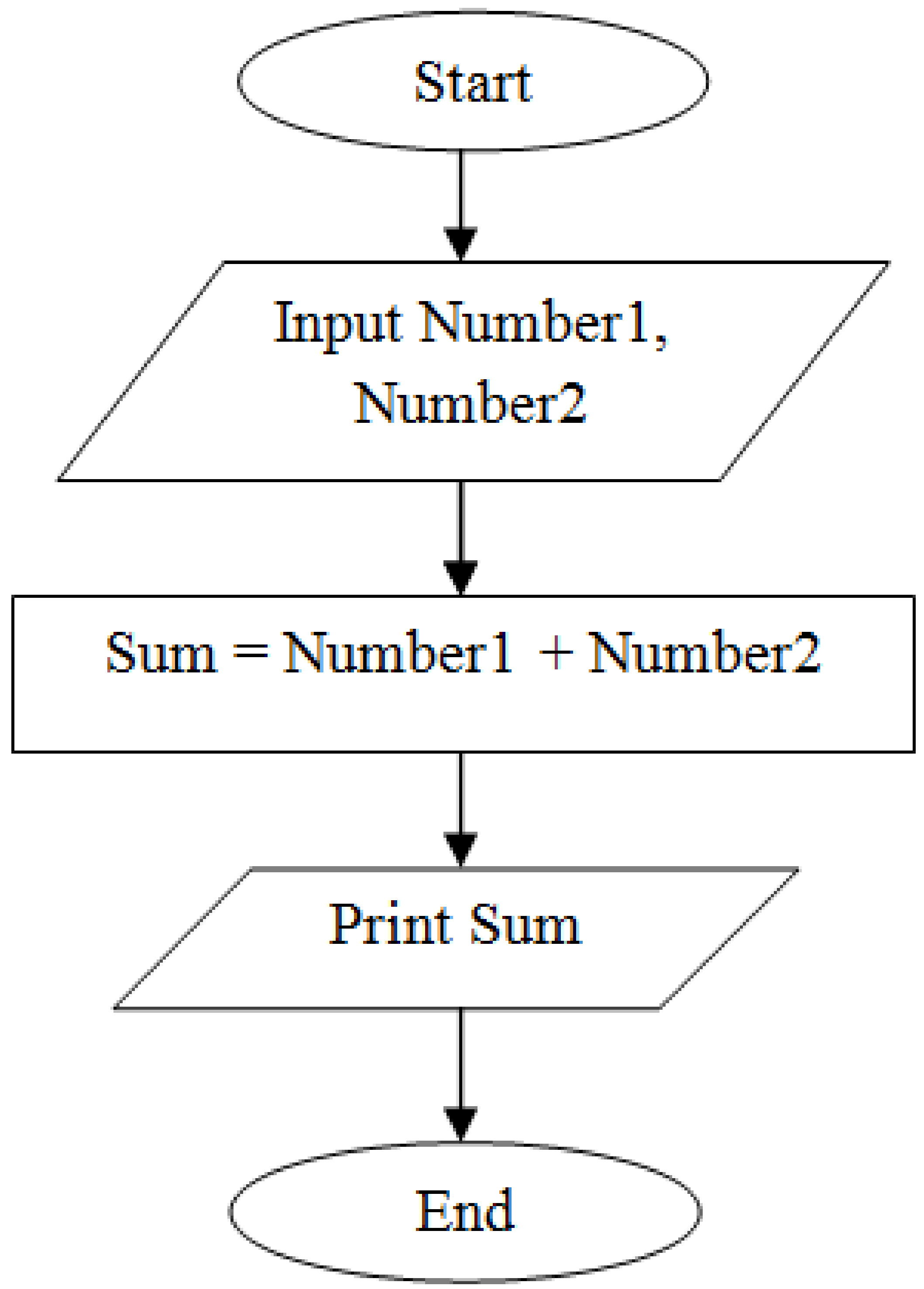
Input: Two numbers num1, num2

Output: Sum of two numbers

1. **sum = num1 + num2**
2. **return sum**

```
void main()
{
    int num1, num2, sum;
    num1 = 5;
}
```





K3

Program

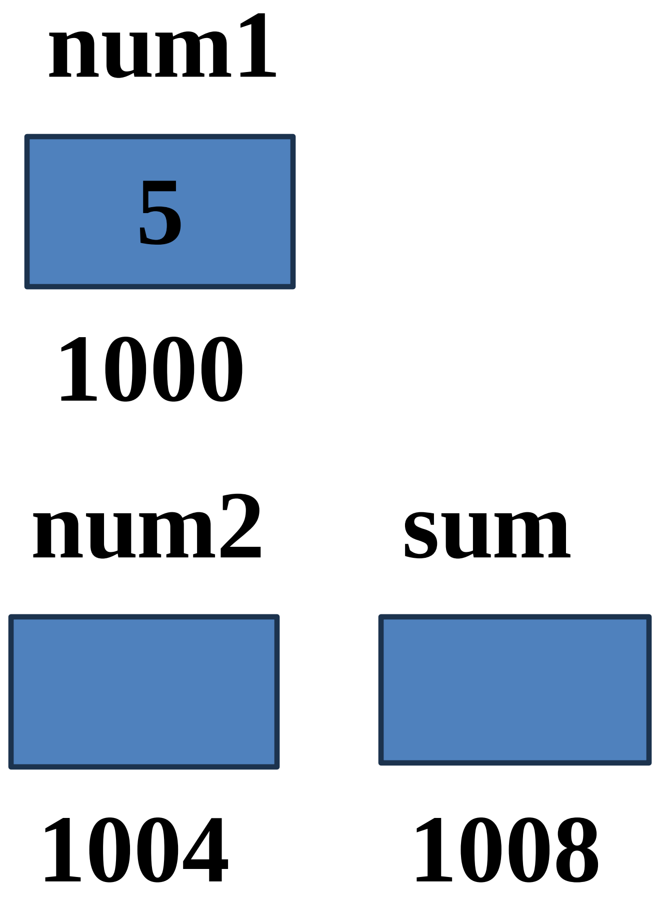
Algorithm 1: Adding two numbers

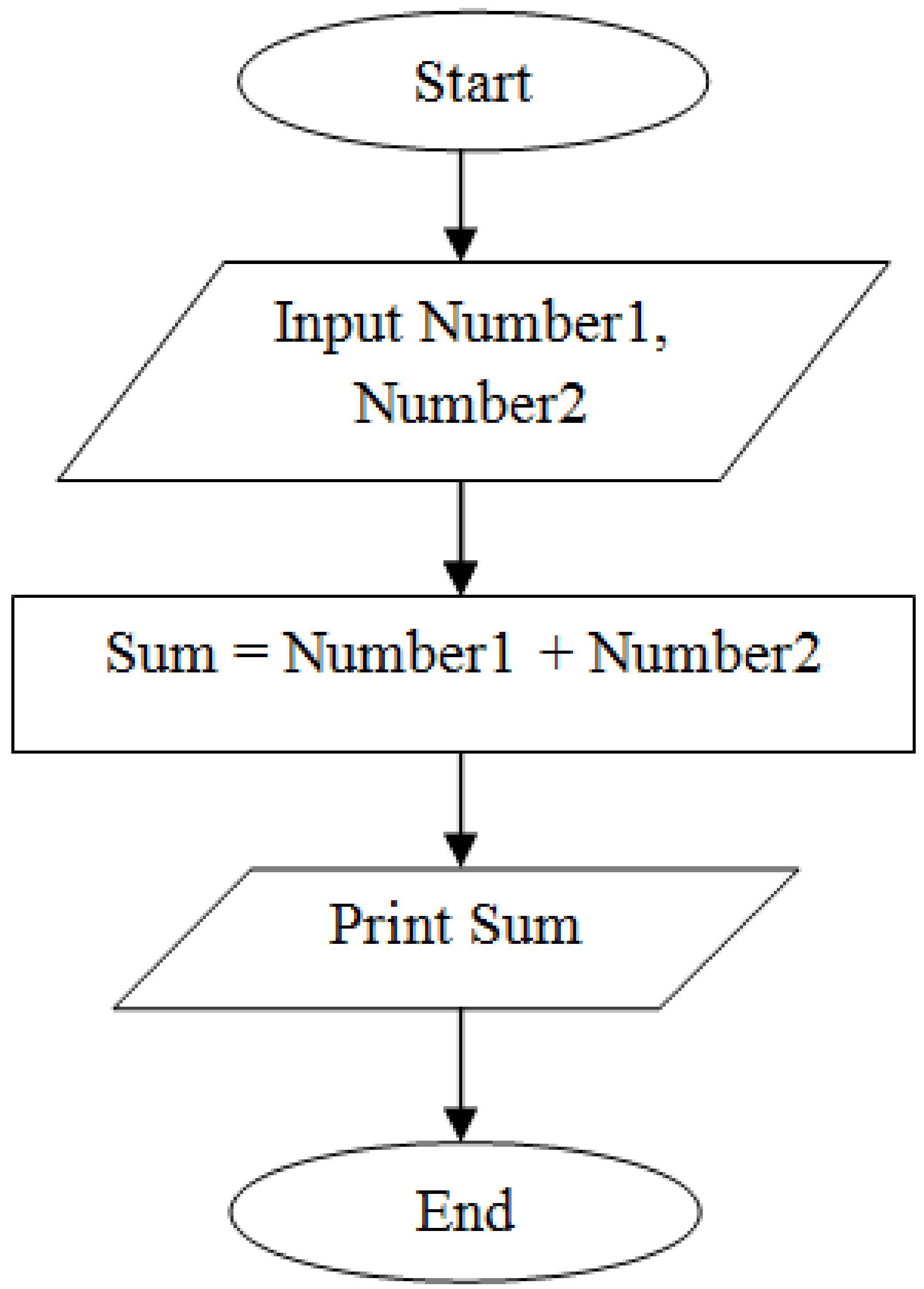
Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.  sum = num1 + num2
2.  return sum
```

```
void main()
{
    int num1, num2, sum;
    num1 = 5;
}
```





K3

Program

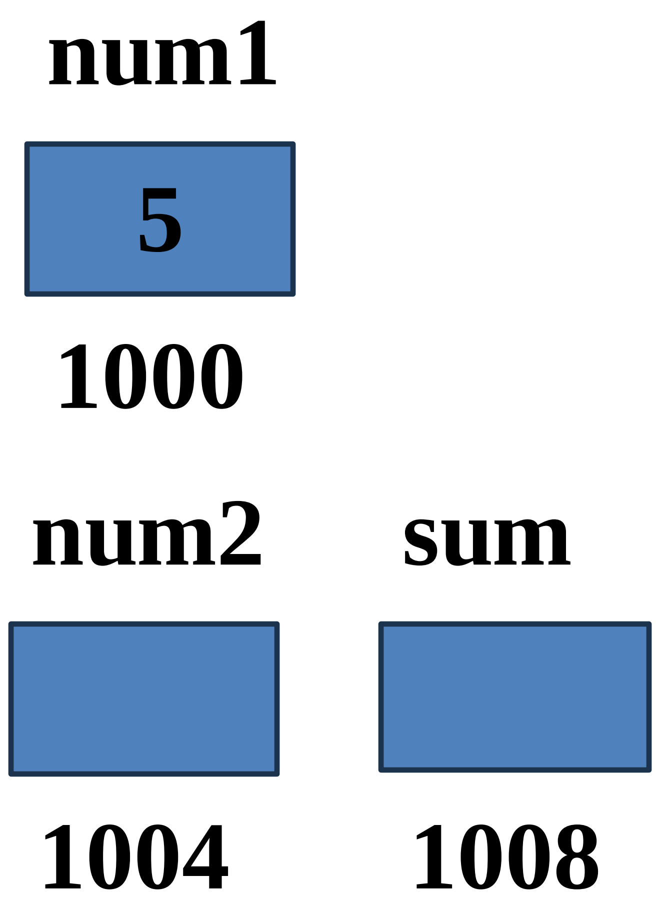
Algorithm 1: Adding two numbers

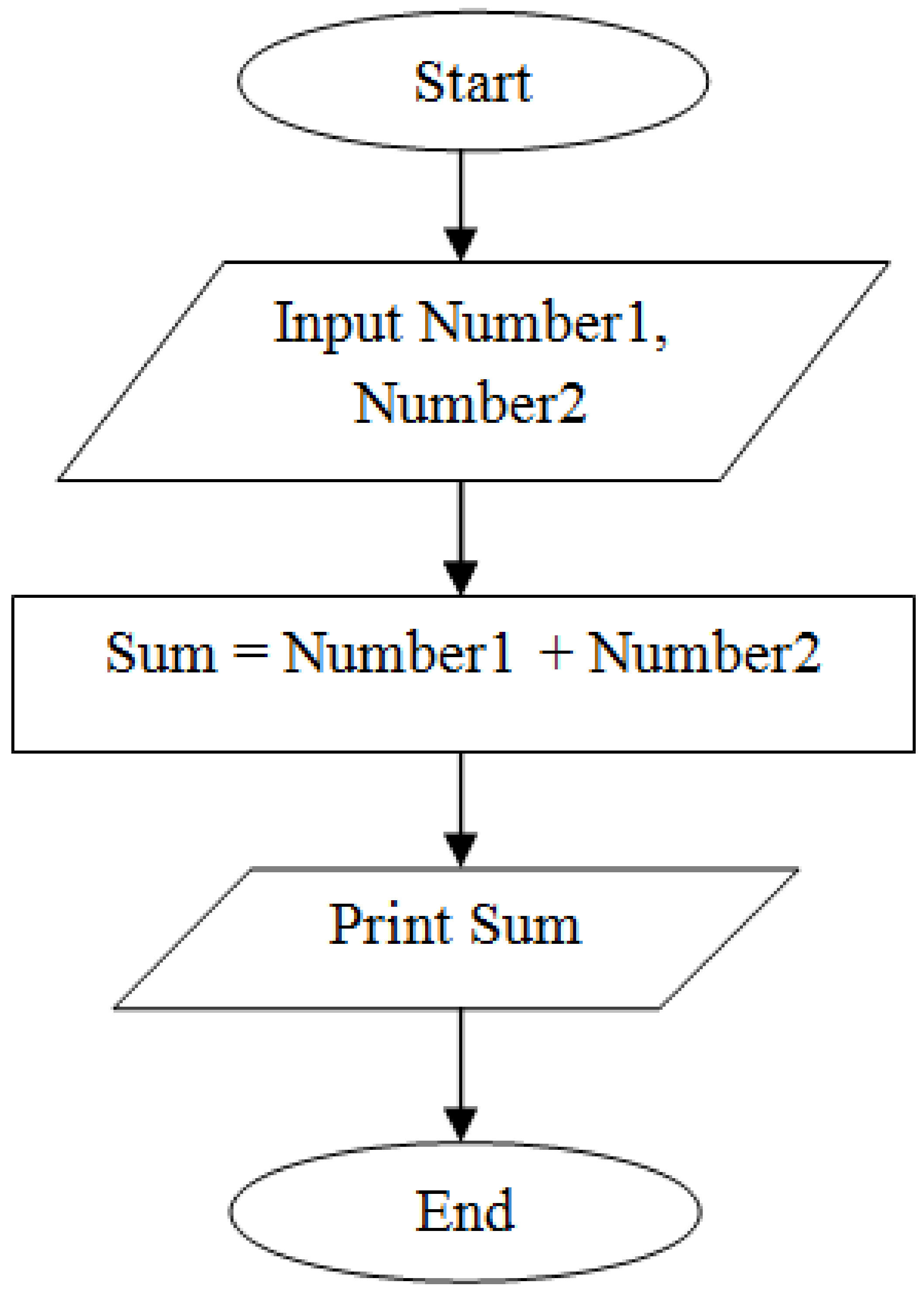
Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.  sum = num1 + num2
2.  return sum
```

```
void main()
{
   int num1, num2, sum;
   num1 = 5;
   num2 = 10;
}
```





K3

Program

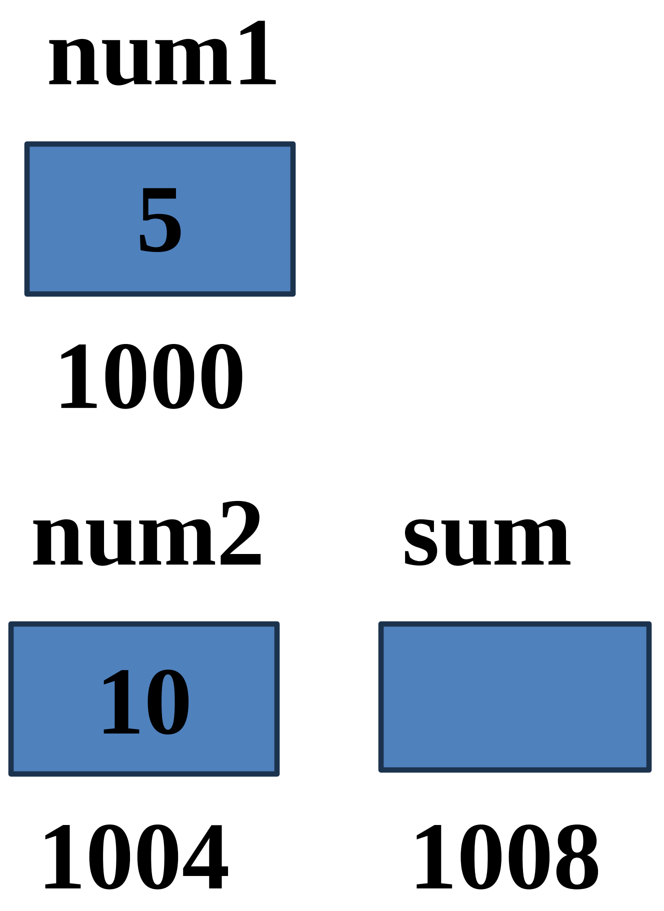
Algorithm 1: Adding two numbers

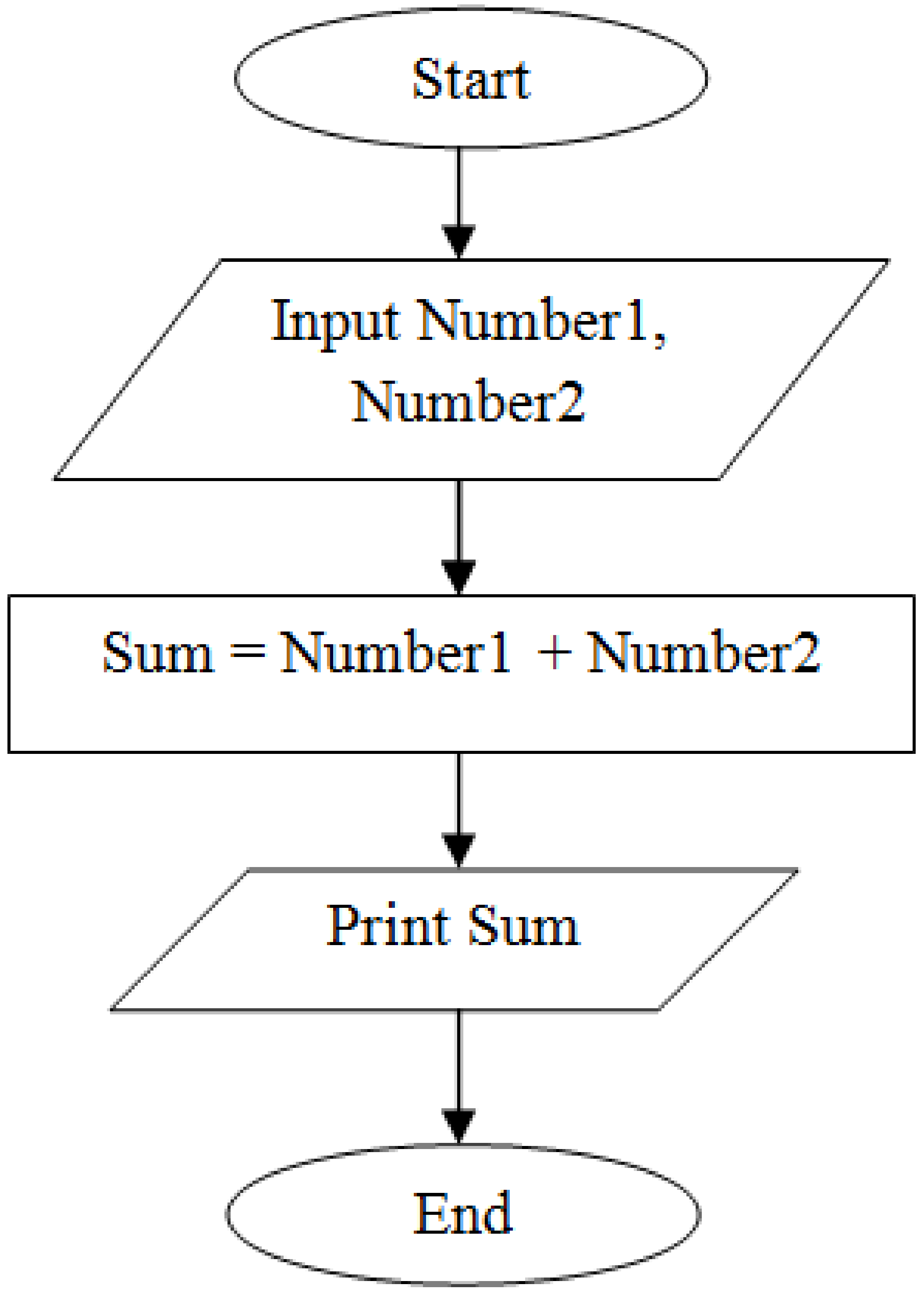
Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.   sum = num1 + num2
2.   return sum
```

```
void main()
{
   int num1, num2, sum;
   num1 = 5;
   num2 = 10;
}
```





K3

Program

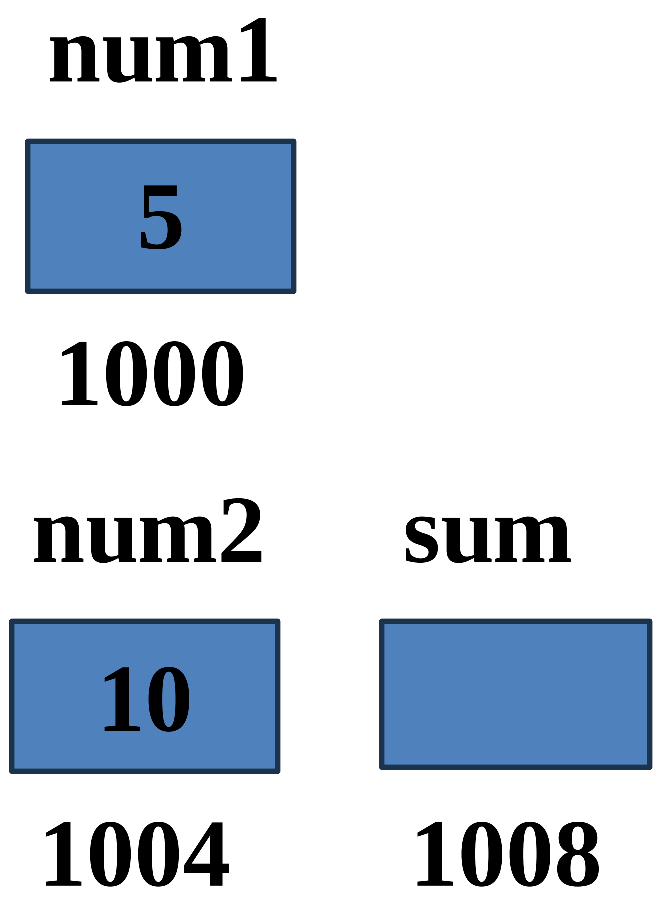
Algorithm 1: Adding two numbers

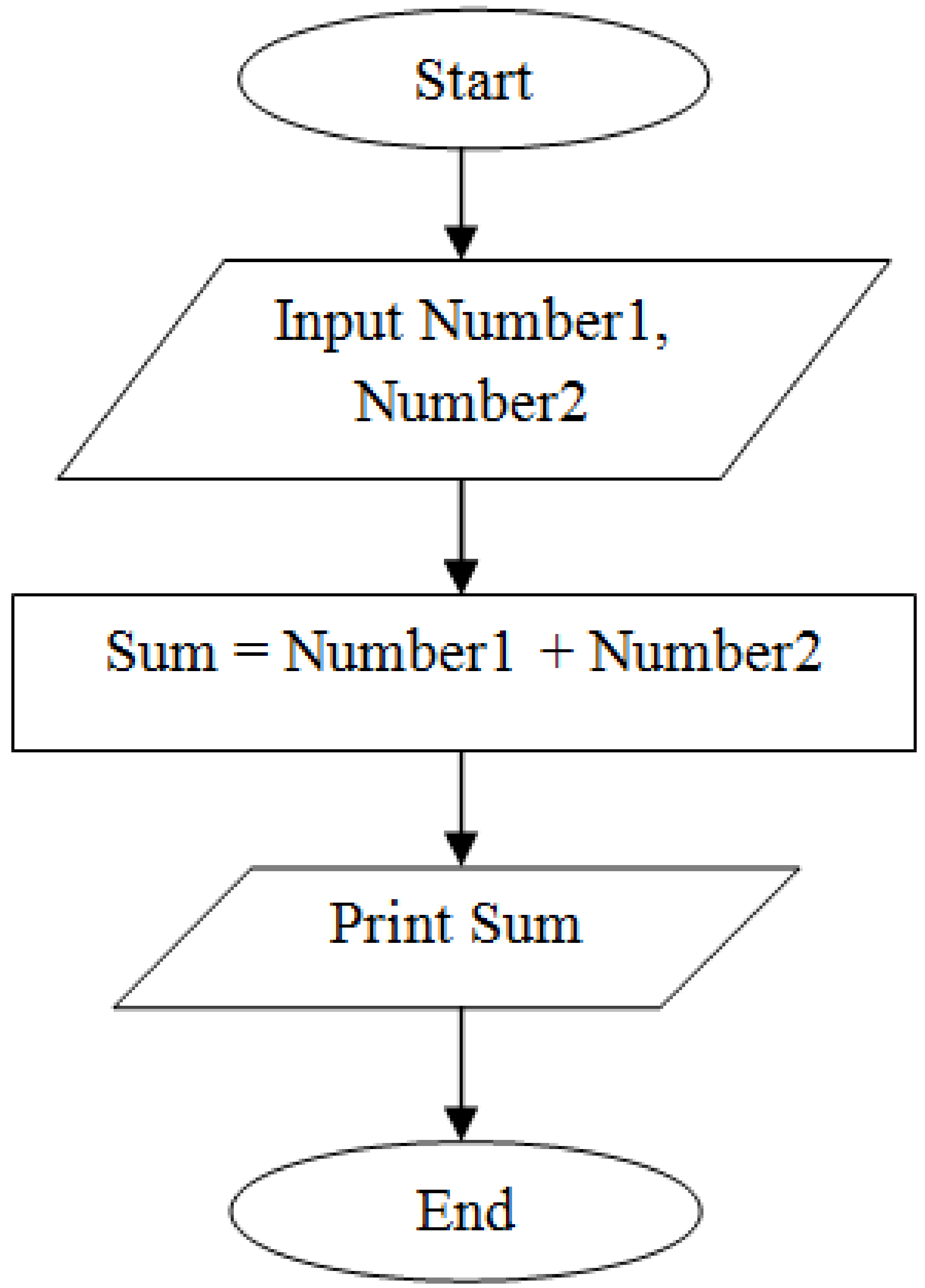
Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.  sum = num1 + num2
2.  return sum
```

```
void main()
{
    int num1, num2, sum;
    num1 = 5;
    num2 = 10;
    sum = num1 + num2;
}
```





K3

Program

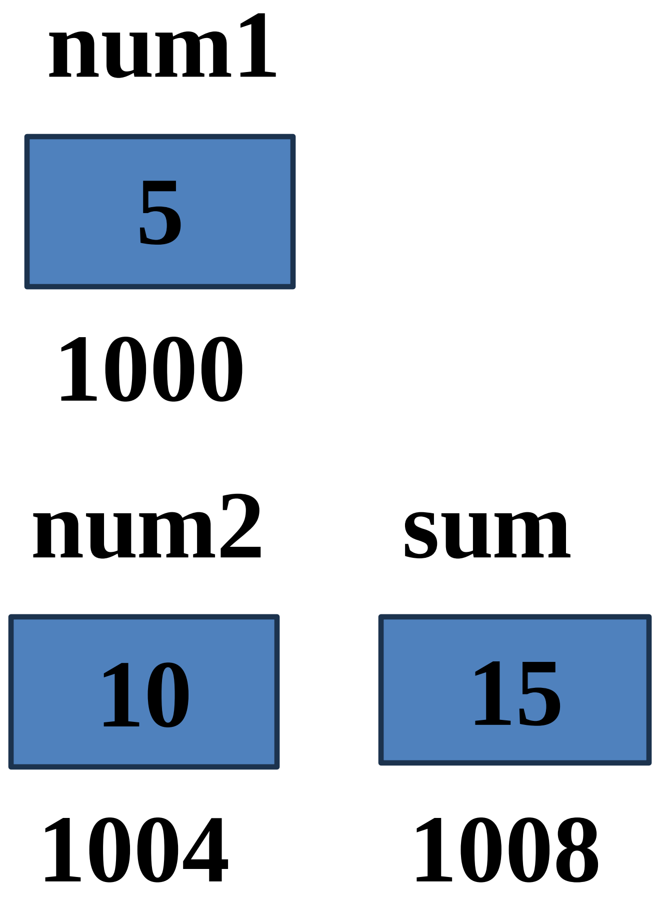
Algorithm 1: Adding two numbers

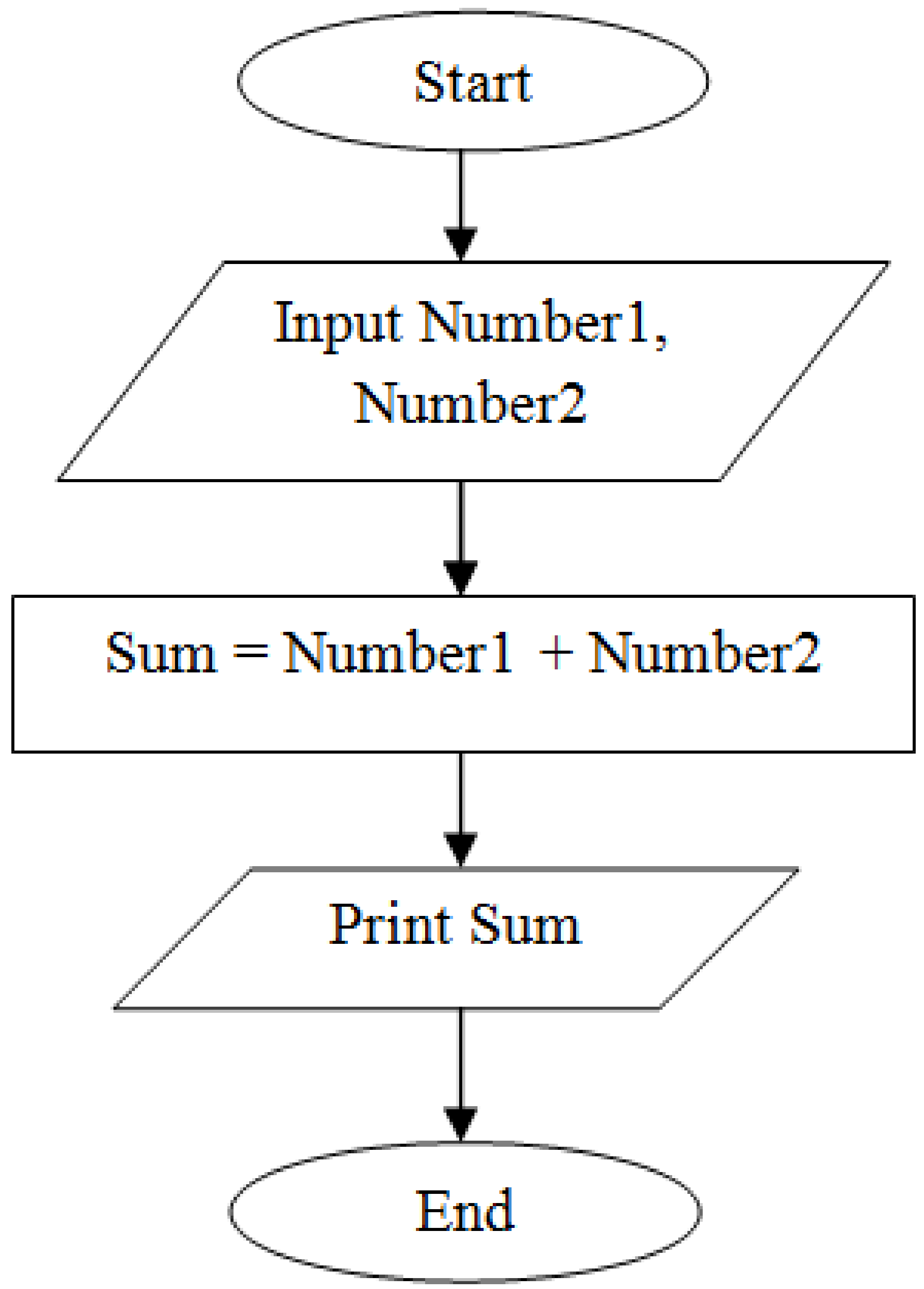
Input: Two numbers num1, num2

Output: Sum of two numbers

```
1.   sum = num1 + num2
2.   return sum
```

```
void main()
{
   int num1, num2, sum;
   num1 = 5;
   num2 = 10;
   sum = num1 + num2;
}
```





K3

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**

K1

Summary

- **Variable Declaration**
- **Program**
- **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