

UST
Faculty of Engineering
First Year [Sem2]

Data Structure and Algorithms

Lecture 3

Primitive Data Structure

Introduction

What is Data?

- Data means facts or values that a computer can store and process.

Examples:

- 25
- 10.5
- 'A'
- "Ahmed"
- True
- These values represent different types of information.

What is a Data Structure?

- A **data structure** is a way of organizing and storing data in a computer so that it can be accessed and processed efficiently.

Example:

Suppose we want to store the marks of five students:

70

85

60

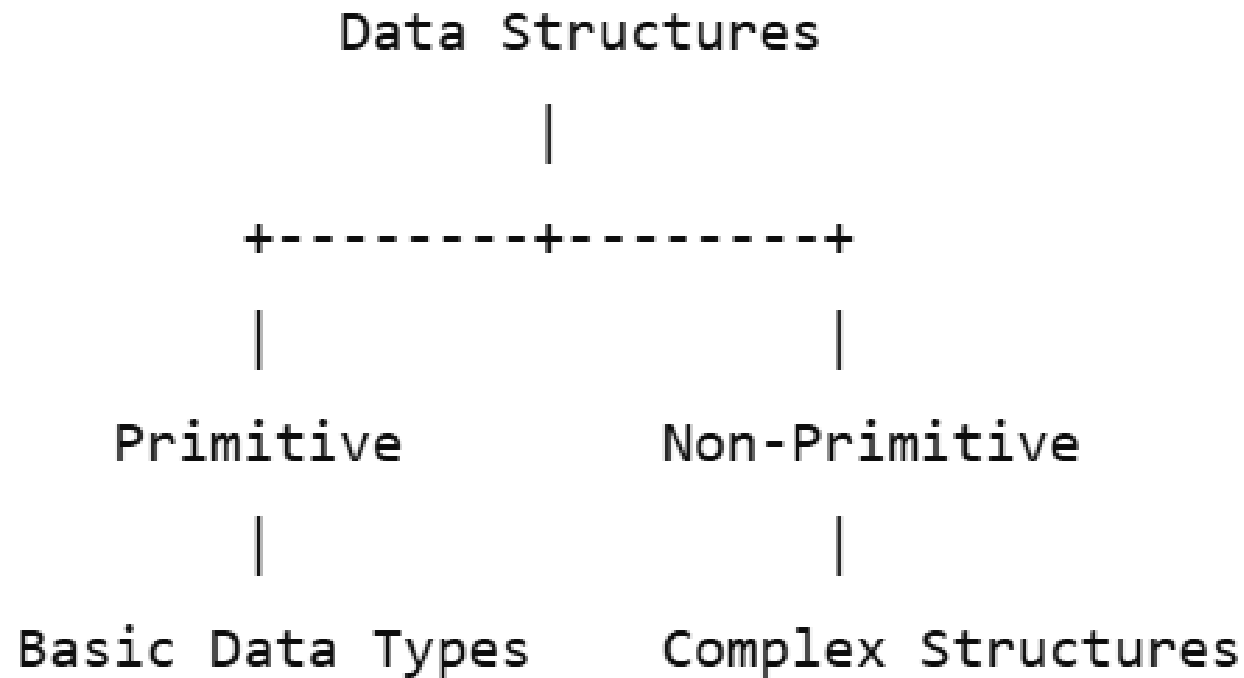
90

75

We need an appropriate structure to store and manage these values.

Classification of Data Structures

- Data structures can generally be divided into two main categories:



Primitive Data Structures

Primitive data structures are the basic data types provided directly by a programming language.

Examples in C++ include:

Int

Float

Double

Char

Bool

They are used to store individual values.

Non-Primitive Data Structures

- Non-primitive data structures are built using primitive data types.

Examples:

- Arrays
- Classes
- Linked Lists
- Stacks
- Queues
- Trees

For example:

- array can contain many values of a primitive type:

int marks[5];

What is a Variable?

- A variable is a named memory location used to store a value.
- Think of a variable as a box with a name.

For example:

- `int age = 20;`
- We can imagine Memory:



- The variable is called age, and its value is 20.

Variable Declaration

Before using a variable, we normally declare it.

Syntax:

data_type variable_name;

Example:

int age;

Here:

- int → data type
- age → variable name

Another example:

float temperature;

Variable Initialization

Initialization means giving a variable its first value.

Example:

int age = 20;

This both declares and initializes the variable.

Another example:

double salary = 2500.50;

Basic Structure of a C++ Program

- A simple C++ program looks like this:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
cout << "Hello World";
```

```
return 0;
```

```
}
```

1. #include <iostream>

- #include <iostream>
- This includes the input/output library.
- It allows us to use:
 - cout
 - cin

1. Using namespace std;

- using namespace std; makes writing C++ programs easier for beginners because it eliminates the need to write std:: every time.

This allows us to write:

Cout

instead of:

std::cout

3. int main()

`int main()`

The `main()` function is where execution of the C++ program begins.

4. cout

```
cout << "Hello World";
```

cout is used to display information or message on the screen.

5. return 0;

return 0;

This indicates that the program finished successfully.

Primitive Data Types

- in C++The most important primitive data types are:

Data Type	Purpose	Example
int	Whole numbers	25
float	Decimal numbers	10.5
double	More precise decimal numbers	25.6789
char	One character	'A'
bool	True/False	true

Integer Data Type – int

The int data type is used to store whole numbers.

Examples:

```
int age = 20;
```

```
int students = 50;
```

```
int temperature = -5;
```

Integers can be:

- **Positive**
- **Negative**
- **Zero**

Example 1:

```
#include <iostream>
using namespace std;
int main(){
int age = 20;
cout << age;
return 0;
}
```

Output
20

Example 2:

```
#include <iostream>
using namespace std;
int main(){
int age = 20;
cout << "Student Age is "<<age;
return 0;
}
```

Output

Student Age is 20

Floating-Point Data Type – float

float is used to store numbers containing decimal values.

Examples:

float temperature = 36.5;

float height = 1.75;

float weight = 65.5;

float price = 25.99;

Example 3:

```
#include <iostream>
using namespace std;
int main(){
float temperature = 36.5;
cout << "Temperature = " << temperature;
return 0;
}
```

Output

Temperature = 36.5

double Data Type – double

double is also used for decimal numbers.

The main difference is that double normally provides more precision than float.

Example:

```
double pi = 3.1415926535;
```

Example 4

```
#include <iostream>
using namespace std;
int main(){
double pi = 3.1415926535;
cout << "Pi = " << pi;
return 0;
}
```

Output

Pi = 3.1415926535

Float vs Double

Feature	float	double
Stores	Decimal numbers	Decimal numbers
Precision	Lower	Higher
Memory	Usually smaller	Usually larger
Common use	Simple decimal values	More precise calculations

Character Data Type – char

The char data type stores one character.

Examples:

```
char grade = 'A';
```

```
char gender = 'M';
```

```
char symbol = '#';
```

A character is written using single quotation marks:

'A'

not:

"A"

Example 5

```
#include <iostream>
using namespace std;
int main(){
    char grade = 'A';
    cout << "Grade = " << grade;
    return 0;
}
```

Output

Grade = A

Important Difference: char and String

A char stores one character:

char letter = 'A';

A string stores multiple characters:

string name = "Ahmed";

For example:

'A' → char

"Ahmed" → string

Remember:

Single quotes → character

Double quotes → text/string

Boolean Data Type – bool

The bool data type stores a logical value.
It can have two possible values:

True

False

Example:

bool passed = true;

bool raining = false;

Example

```
#include <iostream>
using namespace std;
int main(){
    bool passed = true;
    cout << passed;
    return 0;
}
```

Output

1

By default, C++ displays:

true → 1

false → 0

Displaying true and false

We can use: **boolalpha**

Example:

```
#include <iostream>
using namespace std;
int main(){
    bool passed = true;
    cout << boolalpha << passed;
    return 0;
}
```

Output

true