Week# 2
Problem solving techniques

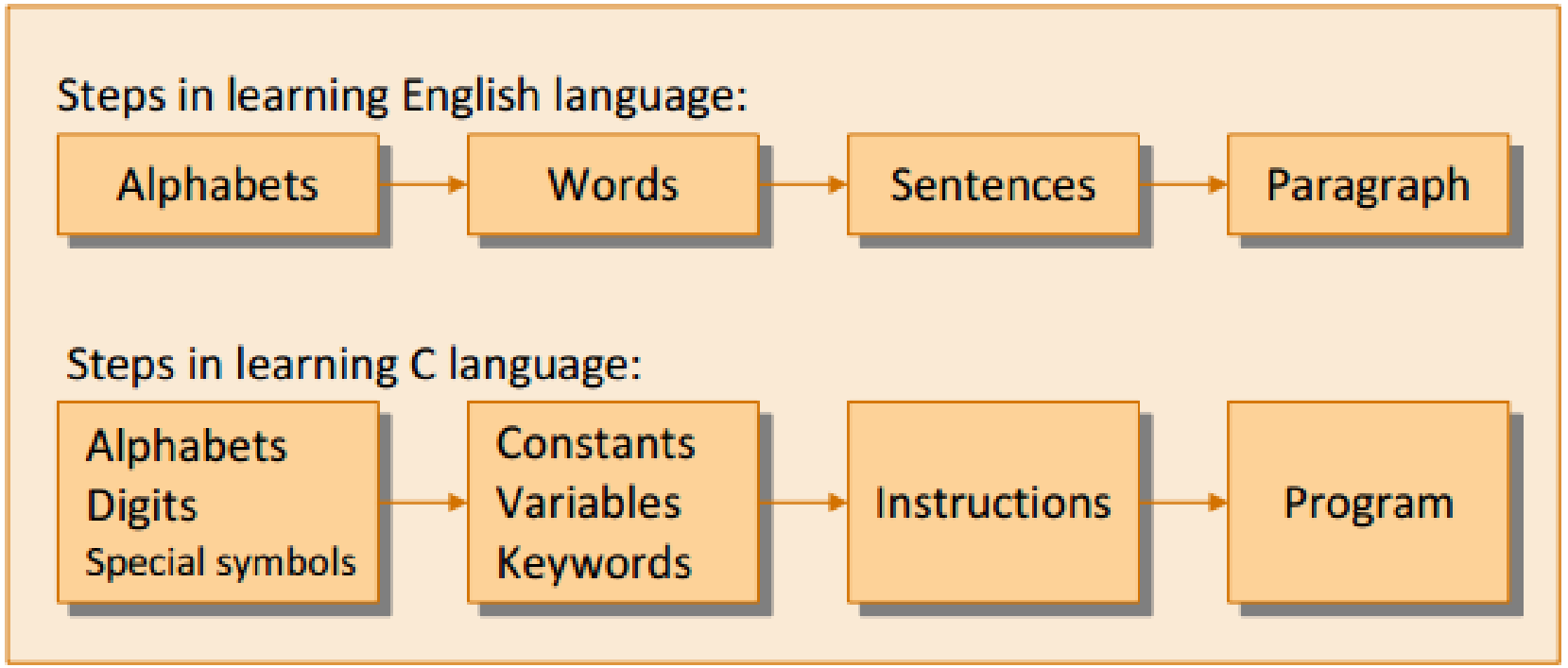
Dr Taimur Ahmed

Department of IT & CS, PAF-IAST

Lecture# 4
Delving in C++ programming: Variables, Data types, Constants

Delving in C++

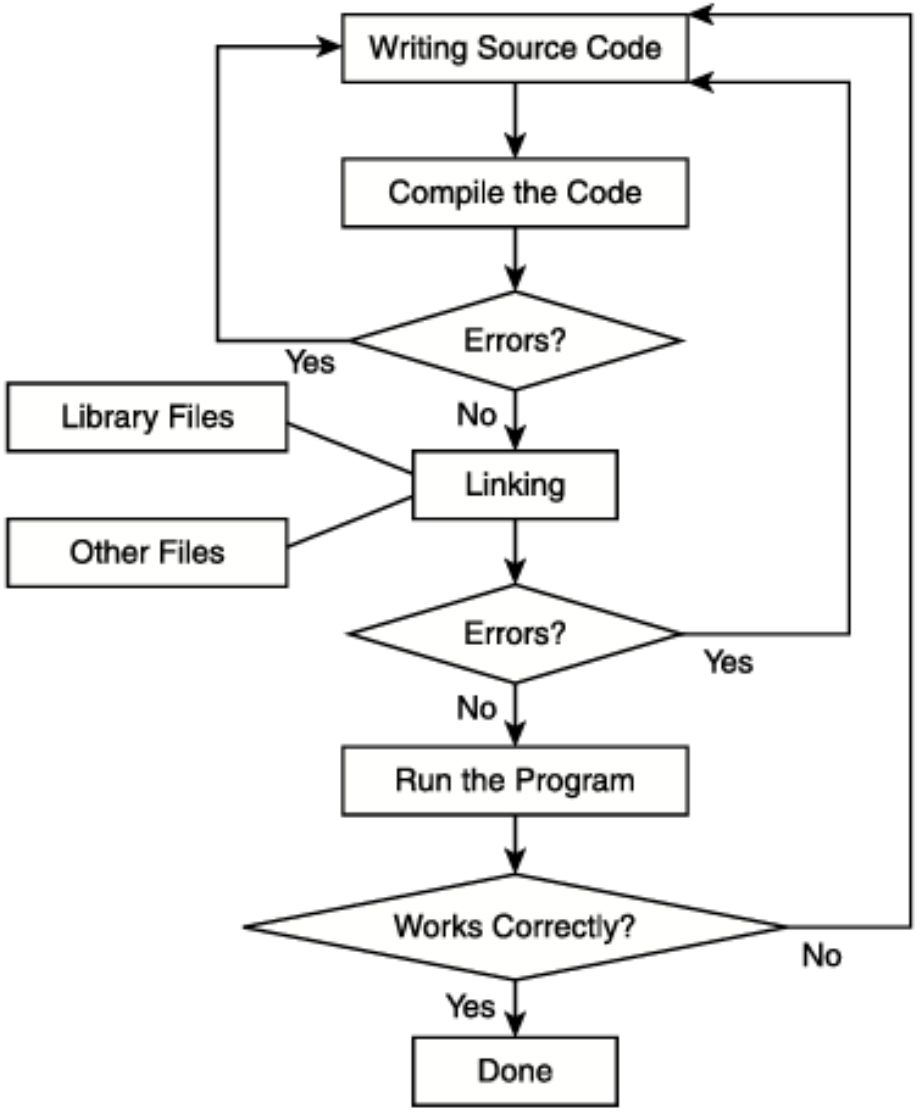
- Learning any **Computer language** is like learning **English language**



- **Character** set in C/C++

Alphabets	A, B, ..., Y, Z a, b, ..., y, z
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Special symbols	~ ‘ ! @ # % ^ & * ( ) _ - + = | \ { } [ ] : ; " ' < > , . ? / $

Program Execution Cycle



Variables & Literals

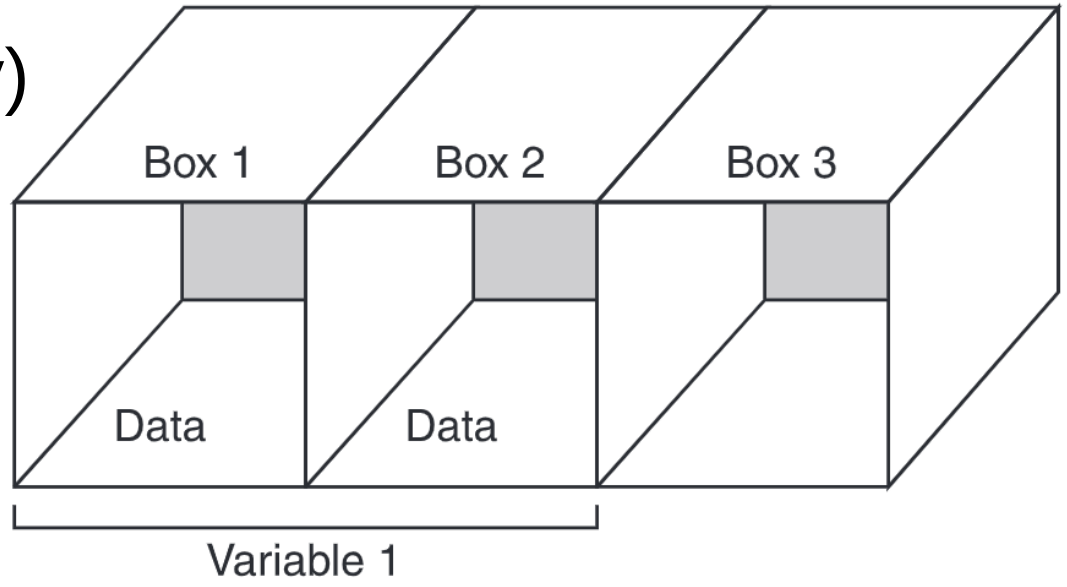
Variables

- **Variable**: a storage location in memory
  - Has a **name and a type** of data it can hold
  - Must be defined before it can be used:

```
int item;
```

- Program use **RAM** (temporary memory)

Boxes represent memory and each box stores 1 byte of data



Variables

- A variable name should **represent the purpose** of the variable. For example:

```
itemsOrdered
```

  - The first character of an identifier must be an **alphabetic character or/and an underscore** (_)
  - After the first character you may use **alphabetic characters, numbers, or underscore characters**
  - Upper and lowercase characters are **distinct**

Variables

- Valid and Invalid **variable names** (also called **Identifiers**)

Identifier	Status	Reason
`totalSales`	Valid	
`total_Sales`	Valid	
`total.Sales`	Invalid	Cannot contain . (dot)
`4thQtrSales`	Invalid	Cannot begin with digit
`totalSale$`	Invalid	Cannot contain special char $

Literals

- **Literal** is a **value stored in memory** when written in a program
  - Literals can be of different data types

```
1  // This program has literals and a variable.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int apples;
8
9     apples = 20;
10    cout << "Today we sold " << apples << " bushels of apples.\n";
11    return 0;
12 }
```

20 is an **integer literal**

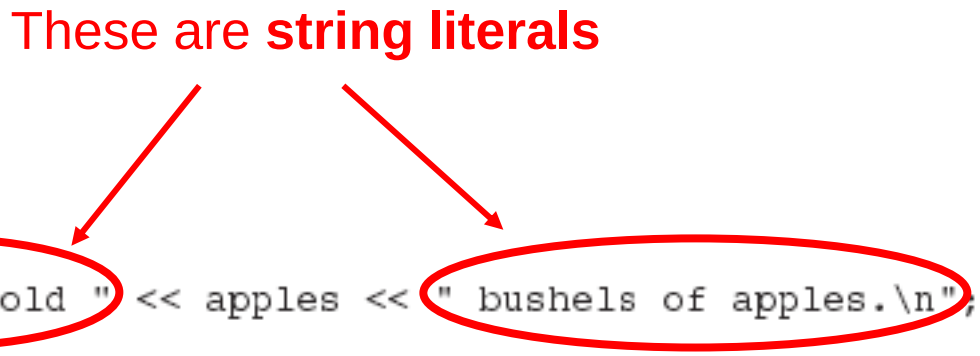
Program Output

```
Today we sold 20 bushels of apples.
```

Literals

❑ String Literals

```
1  // This program has literals and a variable.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int apples;
8
9     apples = 20;
10    cout << "Today we sold " << apples << " bushels of apples.\n";
11    return 0;
12 }
```



Program Output

```
Today we sold 20 bushels of apples.
```

Data Types

`bool, int, char, float`

Data Types

- Data type defines **nature of data** to be store in memory (assigned to a variable)

Type	Keyword
Boolean	`bool`
Character	`char`
Integer	`int`
Floating point	`float`
Double floating point	`double`
Valueless	`void`
Wide character	`wchar_t`

`int` – Integer Data Types

- **Integer variables** can hold **whole numbers**

Data Type	Size	Range
`short`	2 bytes	–32,768 to +32,767
`unsigned short`	2 bytes	0 to +65,535
`int`	4 bytes	–2,147,483,648 to +2,147,483,647
`unsigned int`	4 bytes	0 to 4,294,967,295
`long`	4 bytes	–2,147,483,648 to +2,147,483,647
`unsigned long`	4 bytes	0 to 4,294,967,295

Integer Literals

- An **integer literal** is an **integer value** that is typed in program

```
1  // This program has variables of several of the integer types.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int checking;
8     unsigned int miles;
9     long days;
10
11    checking = -20;
12    miles = 4276;
13    days = 189000;
14    cout << "We have made a long journey of " << miles;
15    cout << " miles.\n";
16    cout << "Our checking account balance is " << checking;
17    cout << "\nAbout " << days << " days ago Columbus ";
18    cout << "stood on this spot.\n";
19    return 0;
20 }
```

`char` – Character Data Type

- Holds **characters or very small integer** values – usually **1 byte** of memory
- **Numeric value** of character is stored in memory

CODE:

```
char letter;
letter = 'C';
```

MEMORY:

`string` – Character Strings

- A **series of characters** in consecutive memory locations:

  `"Hello"`

- Stored with the **null terminator**, `\0`, at the end:
- Comprised of the characters between the " "

`string` – Character Strings

- Special data type supports working with strings

```
#include <string>
```

- Can define `string` variables in programs:

```
string firstName, lastName;
```

- Can receive values with assignment operator:

```
firstName = "George";
lastName = "Washington";
```

- Can be displayed via `cout`

```
cout << firstName << " " << lastName;
```

`string` – Character Strings

```
1   // This program demonstrates the string class.
2   #include <iostream>
3   #include <string> // Required for the string class.
4   using namespace std;
5
6   int main()
7   {
8      string movieTitle;
9
10     movieTitle = "Wheels of Fury";
11     cout << "My favorite movie is " << movieTitle << endl;
12     return 0;
13  }
```

Program Output

```
My favorite movie is Wheels of Fury
```

`float` – Data Type

- The **floating-point** data types are:
  - `float`
  - `double`
  - `long double`

- They can hold **real numbers** such as:
  - `12.45`
  - `-3.8`

Table 2-8 Floating Point Data Types on PCs

Data Type	Key Word	Description
Single precision	`float`	4 bytes. Numbers between ±3.4E-38 and ±3.4E38
Double precision	`double`	8 bytes. Numbers between ±1.7E-308 and ±1.7E308
Long double precision	`long double*`	8 bytes. Numbers between ±1.7E-308 and ±1.7E308

`bool` – Boolean Data Type

- Represents values that are `true` or `false`
- `false` is represented by 0, `true` by 1
- `bool` variables are stored

  as **small integers**

```
1  // This program demonstrates boolean variables.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     bool boolValue;
8
9     boolValue = true;
10    cout << boolValue << endl;
11    boolValue = false;
12    cout << boolValue << endl;
13    return 0;
14 }
```

Program Output

```
1
0
```

`Sizeof` – Determining the size of a Data Type

- The `sizeof()` operator gives the **size of any data type or variable**

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Size of char : " << sizeof(char) << endl;
    cout << "Size of int : " << sizeof(int) << endl;
    cout << "Size of short int : " << sizeof(short int) << endl;
    cout << "Size of long int : " << sizeof(long int) << endl;
    cout << "Size of float : " << sizeof(float) << endl;
    cout << "Size of double : " << sizeof(double) << endl;
    cout << "Size of wchar_t : " << sizeof(wchar_t) << endl;

    return 0;
}
```

Program Output

```
Size of char : 1
Size of int : 4
Size of short int : 2
Size of long int : 4
Size of float : 4
Size of double : 8
Size of wchar_t : 4
```

Scope

Scope

- **Scope of a variable**: the **part of the program** in which the variable can be **accessed**
- A variable cannot be used before it is defined

```
1  // This program can't find its variable.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << value; // ERROR! value not defined yet!
8
9     int value = 100;
10    return 0;
11 }
```

Constants

`cont` – Named Constants

- **Named constant** (constant variable) comprise of values which **cannot be changed** during program execution
- Used for representing **constant values** with descriptive names:

```
const double TAX_RATE = 0.0675;
const int NUM_STATES = 50;
```

- Often named in **uppercase letters**

cont – Named Constants

```
1  // This program calculates the circumference of a circle.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     // Constants
8     const double PI = 3.14159;
9     const double DIAMETER = 10.0;
10
11    // Variable to hold the circumference
12    double circumference;
13
14    // Calculate the circumference.
15    circumference = PI * DIAMETER;
16
17    // Display the circumference.
18    cout << "The circumference is: " << circumference << endl;
19    return 0;
20 }
```

Program Output

```
The circumference is: 31.4159
```