



# Introduction to C++

Lecture-2

Awesh Islam

```
<meta name="description" content="Lastlings official website">
<meta name="author" content="Danny Meneses">

<title>LASTLING S</title>

<link rel="icon" href="img/icons/icon.png">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Ubuntu">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/>
<link rel="stylesheet" href="css/style.css">

<body>
  <div id="main">
    <div id="header">
      <a href="#">Home Page </a>
      <a href="#" class="active">Home </a>
      
      <source src="vid/urgesHD.mp4" type="video/mp4">
    </div>
    <div id="introDetails">
      <h1>LASTLING S</h1>
      <h2>VERSES EP - OUT NOW ON ITUNES & SPOTIFY</h2>
      <img alt="Verses EP album art" style="width: 100%; height: 100%; object-fit: cover;">
    </div>
    <div id="social">
      <a href="https://open.spotify.com/artist/0M7QzubB1PrgqfLdRwIe" target="_blank">SPOTIFY</a>
      <a href="https://soundcloud.com/lastlingslastlings/" target="_blank">SOUND CLOUD</a>
      <a href="https://www.instagram.com/lastlingslastlings/" target="_blank">INSTAGRAM</a>
      <a href="https://www.facebook.com/lastlingslastlings/" target="_blank">FACEBOOK</a>
      <a href="https://twitter.com/LASTLING_S" target="_blank">TWITTER</a>
    </div>
    <div id="navigation">
      <ul>
        <li><a href="#">Navigation </a></li>
        <li><a href="#">Links </a></li>
        <li><a href="tour.html">TOUR</a></li>
        <li><a href="shop.html">SHOP</a></li>
        <li><a href="contact.html">CONTACT</a></li>
      </ul>
    </div>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  </div>
</body>
```

# std::setprecision

```
● ● ●

#include <iostream>
#include <iomanip> // Required for std::setprecision

int main() {
    double pi = 3.14159265358979;

    std::cout << "Default precision: " << pi << std::endl;
    std::cout << "Precision set to 3: " << std::setprecision(3) << pi << std::endl;
    std::cout << "Precision set to 6: " << std::setprecision(6) << pi << std::endl;

    return 0;
}
```

# std::fixed std::scientific

```
#include <iostream>
#include <iomanip> // Required for std::fixed and std::scientific

int main() {
    double num = 12345.6789;

    std::cout << "Fixed-point notation: " << std::fixed << num << std::endl;
    std::cout << "Scientific notation: " << std::scientific << num << std::endl;

    return 0;
}
```

# std::fixed with std::setprecision

```
● ● ●

#include <iostream>
#include <iomanip> // Required for std::fixed and std::setprecision

int main() {
    double num = 123.456789;

    std::cout << "Default (without std::fixed): " << num << std::endl;

    // Use std::fixed to enforce fixed-point notation
    std::cout << "With std::fixed and precision 2: " << std::fixed << std::setprecision(2) << num << std::endl;
    std::cout << "With std::fixed and precision 5: " << std::fixed << std::setprecision(5) << num << std::endl;

    return 0;
}
```

# std::setw

```
● ● ●

#include <iostream>
#include <iomanip> // Required for std::setw

int main() {
    std::cout << std::setw(10) << "Column1" << std::setw(10) << "Column2" << std::endl;
    std::cout << std::setw(10) << 123 << std::setw(10) << 4567 << std::endl;
    std::cout << std::setw(10) << 89 << std::setw(10) << 12 << std::endl;

    return 0;
}
```

# std::setfill

```
● ● ●

#include <iostream>
#include <iomanip> // Required for std::setfill

int main() {
    std::cout << std::setfill('0') << std::setw(5) << 42 << std::endl; // Outputs 00042
    std::cout << std::setfill('-') << std::setw(8) << 123 << std::endl; // Outputs -----123

    return 0;
}
```

std::left std::right

# std::dec std::hex std::oct

```
#include <iostream>

int main() {
    int num = 255;

    std::cout << "Decimal: " << std::dec << num << std::endl;
    std::cout << "Hexadecimal: " << std::hex << num << std::endl;
    std::cout << "Octal: " << std::oct << num << std::endl;

    return 0;
}
```

# Problem



**Problem:** "Formatting Sales Report"

**Statement:** "Awesh is working as a software developer for a small retail company. The company wants to generate a sales report that displays the sales figures for different items in a well-formatted table. Awesh needs to ensure that the prices are aligned properly, with exactly two decimal places, and that all monetary values are prefixed with a dollar sign. Additionally, the report should skip any unnecessary whitespace before user inputs. Awesh decides to use C++ output manipulators to achieve this."

"Write a C++ program that takes the names and prices of three items from the user and then prints a formatted sales report. The prices should be displayed with exactly two decimal places, prefixed with a dollar sign, and aligned in a tabular format."

**Input:**

#The names of three items (each name may contain spaces).

#The prices of the three items as floating-point numbers.

**Output:**

#A table displaying the names and prices of the items, with the prices aligned to the right, showing exactly two decimal places, and prefixed with a dollar sign.

**Constraints:**

#The maximum length of an item name is 20 characters.

**Test Case:**

Laptop	
999.99	
Wireless Mouse	
25.5	
Keyboard	
49.99	

Item	Price
<hr/>	
Laptop	\$999.99
Wireless Mouse	\$25.50
Keyboard	\$49.99

# Solution

```
#include <iostream>
#include <iomanip> // Required for output manipulators

int main() {
    // Declare variables for item names and prices
    std::string item1, item2, item3;
    double price1, price2, price3;

    // Input item names and prices
    std::cout << "Enter the name of the first item: ";
    std::getline(std::cin >> std::ws, item1); // std::ws ignores leading whitespaces
    std::cout << "Enter the price of the first item: ";
    std::cin >> price1;

    std::cout << "Enter the name of the second item: ";
    std::getline(std::cin >> std::ws, item2); // std::ws ignores leading whitespaces
    std::cout << "Enter the price of the second item: ";
    std::cin >> price2;

    std::cout << "Enter the name of the third item: ";
    std::getline(std::cin >> std::ws, item3); // std::ws ignores leading whitespaces
    std::cout << "Enter the price of the third item: ";
    std::cin >> price3;

    // Output the formatted table header
    std::cout << std::left << std::setw(20) << "Item" << " | "
        << std::right << std::setw(8) << "Price" << std::endl;
    std::cout << "-----+-----" << std::endl;

    // Output the formatted table rows
    std::cout << std::left << std::setw(20) << item1 << " | "
        << std::right << std::setw(8) << std::fixed << std::setprecision(2) << "$" << price1 << std::endl;

    std::cout << std::left << std::setw(20) << item2 << " | "
        << std::right << std::setw(8) << std::fixed << std::setprecision(2) << "$" << price2 << std::endl;

    std::cout << std::left << std::setw(20) << item3 << " | "
        << std::right << std::setw(8) << std::fixed << std::setprecision(2) << "$" << price3 << std::endl;

    return 0;
}
```

# Const variables

```
● ● ●

#include <iostream>

int main() {
    const int max_students = 30;      // Constant integer variable
    const double pi = 3.14159;        // Constant double variable

    std::cout << "The maximum number of students allowed is: " << max_students << std::endl;
    std::cout << "The value of pi is approximately: " << pi << std::endl;

    // Uncommenting the following lines will cause a compilation error
    // max_students = 35; // Error: cannot modify a const variable
    // pi = 3.14;         // Error: cannot modify a const variable

    return 0;
}
```

# Inline function

```
● ● ●

#include <iostream>

// Inline function to calculate the square of a number
inline int square(int x) {
    return x * x;
}

int main() {
    int num1 = 5;
    int num2 = 10;

    // Using the inline function to calculate squares
    std::cout << "The square of " << num1 << " is: " << square(num1) << std::endl;
    std::cout << "The square of " << num2 << " is: " << square(num2) << std::endl;

    return 0;
}
```

# std::string



```
#include <iostream>
#include <string>

int main()
{
    std::string name { "Awesh" };
    std::cout << "My name is: " << name << '\n';

    return 0;
}
```

# Benefits of using string

```
#include <iostream>
#include <string>

int main()
{
    std::string name { "Awesh" }; // initialize name with string literal "Alex"
    std::cout << name << '\n';

    name = "Jaber";           // change name to a longer string
    std::cout << name << '\n';

    name = "Emon";           // change name to a shorter string
    std::cout << name << '\n';

    return 0;
}
```

# String Input

```
#include <iostream>
#include <string>

int main()
{
    std::cout << "Enter your full name: ";
    std::string name{};
    std::cin >> name; // this won't work as expected since std::cin breaks on whitespace

    std::cout << "Enter your favorite color: ";
    std::string color{};
    std::cin >> color;

    std::cout << "Your name is " << name << " and your favorite color is " << color << '\n';

    return 0;
}
```

# std::getline()

```
● ● ●

#include <iostream>
#include <string>

int main() {
    std::string fullName;

    std::cout << "Enter your full name: ";
    std::getline(std::cin, fullName); // Reading a full line of input, including spaces

    std::cout << "Hello, " << fullName << "!" << std::endl;

    return 0;
}
```

# Problem of using get line

```
#include <iostream>
#include <string>

int main() {
    int age;
    std::string fullName;

    std::cout << "Enter your age: ";
    std::cin >> age; // Input an integer

    std::cout << "Enter your full name: ";
    std::getline(std::cin, fullName); // Problem: This will read an empty line

    std::cout << "Age: " << age << ", Full Name: " << fullName << std::endl;

    return 0;
}
```

# Input manipulator: std::ws

```
#include <iostream>

int main() {
    std::string str;
    std::cout << "Enter a string with leading spaces: ";
    std::cin >> std::ws; // Skips leading whitespace
    std::getline(std::cin, str);
    std::cout << "You entered: \" " << str << " \" " << std::endl;

    return 0;
}
```

# Solution

```
● ● ●

#include <iostream>
#include <string>

int main()
{
    std::cout << "Pick 1 or 2: ";
    int choice{};
    std::cin >> choice;

    std::cout << "Now enter your name: ";
    std::string name{};
    std::getline(std::cin >> std::ws, name); // note: added std::ws here

    std::cout << "Hello, " << name << ", you picked " << choice << '\n';

    return 0;
}
```

# string.length()

```
● ● ●

#include <iostream>
#include <string>

int main()
{
    std::string name{ "Awesh" };
    std::cout << name << " has " << name.length() << " characters\n";

    return 0;
}
```