



Introduction to C++

Lecture-1

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>
  <a href="#">Home Page </a>
  <a class="home"></a>
  <video poster="versesBg.png" autoplay="true" loop>
    <source src="vid/urgesHD.mp4" type="video/mp4">
  </video>
  <a class="introDetails">
    <h1>LASTLING S</h1>
    <h2>VERSES EP - OUT NOW ON ITUNES & SPOTIFY</h2>
    <h3>RELEASED 10/10/15</h3>
    <a href="https://open.spotify.com/artist/0M7QzubB1PrgqfLdRwIe" target="_blank">SPOTIFY</a>
    <a href="https://soundcloud.com/lastlings" 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>
  </a>
  <ul class="navigation">
    <li class="links">
      <a href="tour.html">TOUR</a>
      <a href="shop.html">SHOP</a>
      <a href="contact.html">CONTACT</a>
    </li>
  </ul>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

```

History of C++

- C++ Was developed by Bjarne Stroustrup at Bell Labs as an extension to C, starting in 1979
- C++ was standardized in 1998 by the ISO committee
- Five major updates to the C++ language (C++11, C++14, C++17, C++20, and C++23) have been made since then, each adding additional functionality.



Your 1st C++ Program

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

Output (`std::cout` & `std::endl`)

```
● ● ●

#include <iostream> // for std::cout and std::endl

int main()
{
    std::cout << "Hi!" << std::endl; // std::endl will cause the
cursor to move to the next line
    std::cout << "My name is Alex." << std::endl;

    return 0;
}
```

Output different variables

```
#include <iostream>

int main() {
    // Declare variables of different data types
    int age = 25;
    double height = 5.9;
    char initial = 'A';

    // Output the different types of data using cout
    std::cout << "Name: " << name << std::endl;          // Output a string
    std::cout << "Initial: " << initial << std::endl;    // Output a character
    std::cout << "Height: " << height << " feet" << std::endl; // Output a double

    return 0; // Indicate that the program ended successfully
}
```

Input (std::cin)

```
#include <iostream> // for std::cout and std::cin

int main()
{
    std::cout << "Enter a number: " // ask user for a number

    int x; // define variable x to hold user input (and value-initialize it)
    std::cin >> x; // get number from keyboard and store it in variable x

    std::cout << "You entered " << x << '\n';
    return 0;
}
```

Input and Output

```
● ● ●

#include <iostream> // for std::cout and std::cin

int main()
{
    std::cout << "Enter two numbers: ";

    int x;
    std::cin >> x;

    int y{};
    std::cin >> y;

    std::cout << "You entered " << x << " and " << y << '\n';

    return 0;
}
```

Comments

```
#include <iostream>

int main() {
    // Declare a variable to store the user's age
    int age;

    // Prompt the user to enter their age
    std::cout << "Enter your age: ";
    std::cin >> age;

    /*
     * Output a message to the user based on their age.
     * The message will include the age they entered.
     */
    std::cout << "You are " << age << " years old." << std::endl;

    return 0; // Indicate that the program ended successfully
}
```

Let's solve a problem



Take a number input from user and double it

Solution

```
#include <iostream>

// preferred version
int main()
{
    std::cout << "Enter an integer: ";

    int num;
    std::cin >> num;

    std::cout << "Double that number is: " << num * 2 << '\n';
    // use an expression to multiply num * 2 at the point where we are going to print it

    return 0;
}
```

List initialiser

- **List Initialisation** in C++ allows you to initialise variables using curly braces '{}':
- Introduced in C++11, it provides a consistent syntax for initializing various types of data.

```
● ● ●  
int x{5};      // Initializes x with the value 5  
double pi{3.14}; // Initializes pi with the value 3.14
```

```
● ● ●  
int arr[3]{1, 2, 3}; // Initializes an array with values 1, 2, 3
```

List Intialiser

```
struct Point {  
    int x, y;  
};  
  
Point p{10, 20}; // Initializes a Point with x = 10, y = 20
```

List Initialiser Advantages



```
int a{2.9}; // Error: narrowing conversion from double to int
```

Name Collisions

```
#include <iostream>

void myFcn(int x)
{
    std::cout << x;
}

void myFcn(int x)
{
    std::cout << 2 * x;
}

int main()
{
    return 0;
}
```

Name Collisions

```
● ○ ●  
#include <iostream>  
  
void myFcn(int x)  
{  
    std::cout << 2 * x;  
}  
  
int main()  
{  
    return 0;  
}
```

main.cpp

```
● ○ ●  
void myFcn(int x)  
{  
    std::cout << x;  
}
```

header.h

How C++ handles name collision?

Answer: namespaces

How we declare namespaces and details will be discussed in a later lecture

```
● ● ●

#include <iostream>

int main()
{
    std::cout << "Hello world!";
    // when we say cout, we mean the cout defined in the
    std namespace
    return 0;
}
```

'using' directive

```
#include <iostream>

using namespace std; // this is a using-directive that
// allows us to access names in the std namespace with no
// namespace prefix

int main()
{
    cout << "Hello world!";
    return 0;
}
```

Problem with the 'using' directive

```
#include <iostream> // imports the declaration of std::cout into the global scope

using namespace std; // makes std::cout accessible as "cout"

int cout() // defines our own "cout" function in the global namespace
{
    return 5;
}

int main()
{
    cout << "Hello, world!"; // Compile error! Which cout do we want here?
    // The one in the std namespace or the one we defined above?

    return 0;
}
```

Pre-processors

#include

```
● ● ●

#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
    return 0;
}
```

New data-type

bool

```
#include <iostream>

int main()
{
    std::cout << true << '\n'; // true evaluates to 1
    std::cout << !true << '\n'; // !true evaluates to 0

    bool b {false};
    std::cout << b << '\n'; // b is false, which evaluates to 0
    std::cout << !b << '\n'; // !b is true, which evaluates to 1
    return 0;
}
```

Std::boolalpha

Controls how booleans are printed

```
#include <iostream>

int main()
{
    std::cout << true << '\n';
    std::cout << false << '\n';

    std::cout << std::boolalpha; // print bools as true or false

    std::cout << true << '\n';
    std::cout << false << '\n';
    return 0;
}
```

How to take bool input?

```
#include <iostream>

int main()
{
    bool b{}; // default initialize to false
    std::cout << "Enter a boolean value: ";
    std::cin >> b;
    std::cout << "You entered: " << b << '\n';

    return 0;
}
```

0 for false

1 for true

We don't enter 'true' or 'false'