



# TYPES OF LITERALS

## C++ BASICS

---

*prepared by:*

Gyro A. Madrona

Electronics Engineer

# TOPIC OUTLINE

Integer Literals

Floating-Point Literals

Character Literals

String Literals

Boolean Literals



# TYPES OF LITERALS



# LITERALS

---

A literal is a notation in the source code that directly represents a fixed, constant value. Unlike variables or expressions, literals are the actual values themselves.



# INTEGER LITERALS

---

Integer literals represent whole numbers. It can be written in decimal, octal, hexadecimal, or binary formats.

Examples:

```
int decimal = 24;
```

```
int octal = 052;
```

```
int hexadecimal = 0x2A;
```

```
int binary = 0b101010;
```



# FLOATING-POINT LITERALS

Examples:

```
double decimal = 5.15;
```

```
//  $6.02 \times 10^{23}$ 
```

```
double exponential = 6.02e23
```

Floating-point literals represent real numbers (numbers with fractional parts). It can be written in decimal or exponential notation.



# CHARACTER LITERALS

---

Character literals represent single characters enclosed in single quote.

Examples:

```
char character = 'A';
```

```
char newline = '\n';
```

```
// hexadecimal representation of 'A'
```

```
char hexChar = '\x41';
```

Escape sequences are used to represent **special characters**. They start with backslash (\) followed by a specific character.



# ESCAPE SEQUENCES

Escape Sequence	Description	Example
<code>\n</code>	Newline	<code>cout &lt;&lt; "Hello\nWorld";</code>
<code>\t</code>	Horizontal tab	<code>cout &lt;&lt; "Name:\tJohn";</code>
<code>\\</code>	Backslash	<code>cout &lt;&lt; "C:\\folder\\file";</code>
<code>\'</code>	Single quote	<code>char c = '\''</code>
<code>\"</code>	Double quote	<code>cout &lt;&lt; "\"Hello\"";</code>
<code>\a</code>	Alert (bell)	<code>cout &lt;&lt; "\a";</code>
<code>\b</code>	Backspace	<code>cout &lt;&lt; "Hello\b";</code>
<code>\f</code>	Form feed (page break)	<code>cout &lt;&lt; "Hello\fWorld";</code>
<code>\r</code>	Carriage return	<code>cout &lt;&lt; "Hello\rWorld";</code>
<code>\v</code>	Vertical tab	<code>cout &lt;&lt; "Hello\vWorld";</code>
<code>\xhh</code>	Hexadecimal	<code>char c = '\x41'; (ASCII 'A')</code>
<code>\ooo</code>	Octal	<code>char c = '\101'; (ASCII 'A')</code>



# STRING LITERALS

---

Examples:

```
string message = "Hello, World!";
```

String literals represent sequences of characters enclosed in double quotes.



# BOOLEAN LITERALS

---

Examples:

```
bool status = true;
```

```
bool status = false;
```

Boolean literals represent true or false values.



## EXERCISE

---

Determine the output of this code snippet:

```
int oct_1 = 07;
```

```
int oct_2 = 01;
```

```
int sum = 00;
```

```
sum = oct_1 + oct_2;
```

```
cout<<oct<<sum;
```

output:

Determine the output of this code snippet:

```
int hex_1 = 0xE;
```

```
int hex_2 = 0x1;
```

```
int sum = 0x0;
```

```
sum = hex_1 + hex_2;
```

```
cout<<uppercase<<hex<<sum;
```

output:



## EXERCISE

---

Determine the output of this code snippet:

```
#include <bitset>

int bin_1 = 0b101;

int bin_2 = 0b001;

int sum = 0b0;

sum = bin_1 + bin_2;

cout<<bitset<3>(sum) ;
```

output:

Determine the output of this code snippet:

```
string name = "Ada Lovelace";

char sex = 'F' ;

int age = 27;

cout<<"Name: \t"<<name<<"\n" ;

cout<<"Sex: \t"<<sex<<"\n" ;

cout<<"Age: \t"<<age ;
```

output:



# LABORATORY

