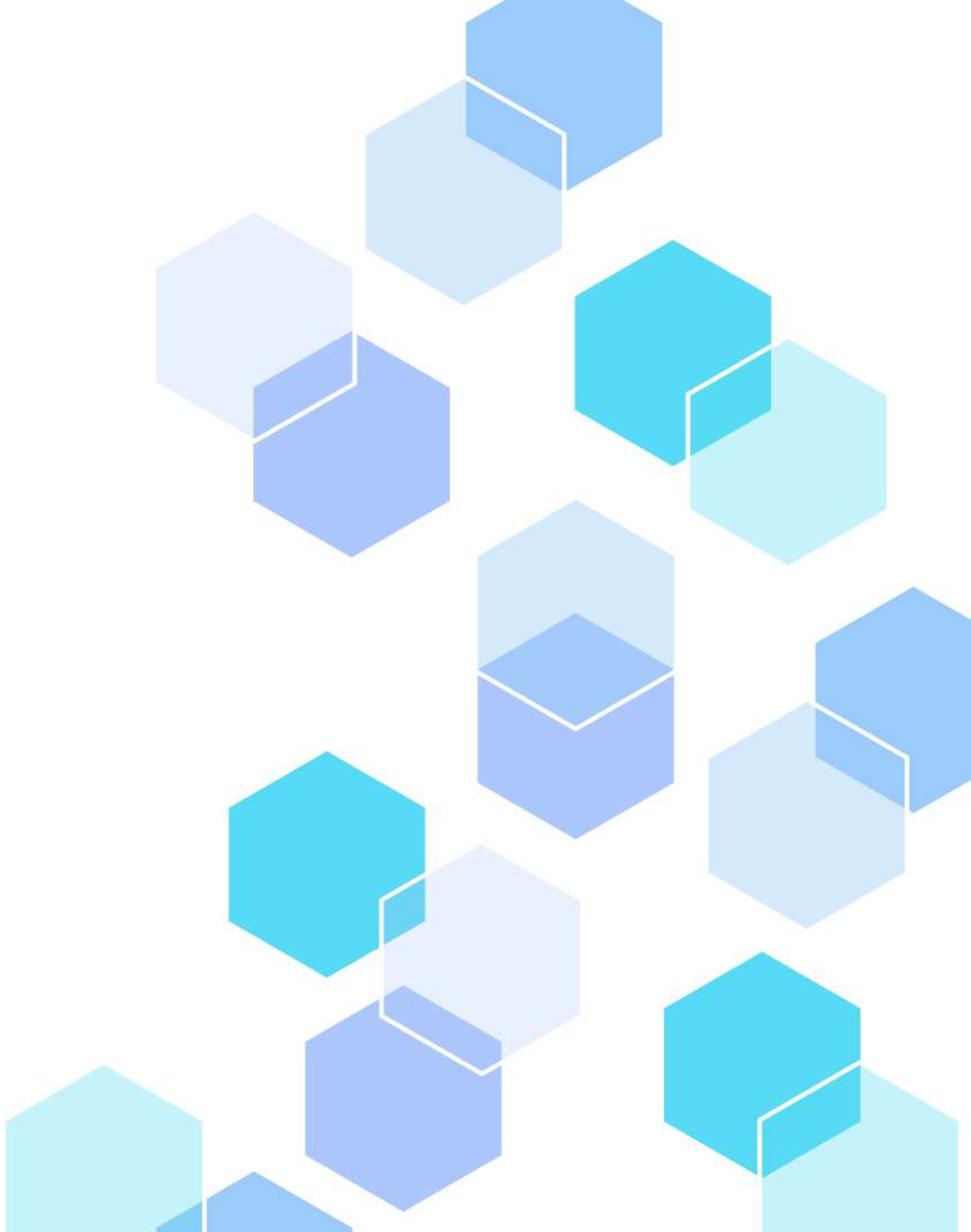
Table of contents

01

Structure

Basic Program Structure of C++



Basic Program Structure of C++

Basic Program Structure of C++

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

int main()
{
  cout << "Hello World";

  return 0;
}
```

Basic Program Structure of C++

Pre-Processor

Library/Header Files

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

int main()
{
    cout << "Hello World";

    return 0;
}
```

Built-in object

Basic Program Structure of C++

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

int main()
{
  cout << "Hello World";

  return 0;
}
```

Library File

```cpp
#include<iostream>
```

Input Output Stream header file

# #

Pre-Processor

This **tells the compiler** that first pre-process (executes) the **process of importing a library/header file.**

Basic Program Structure of C++

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

int main()
{
  cout << "Hello World";

  return 0;
}
```

Namespace

The std is a short form of **standard**, the **std namespace** contains the **built-in classes and declared functions.**

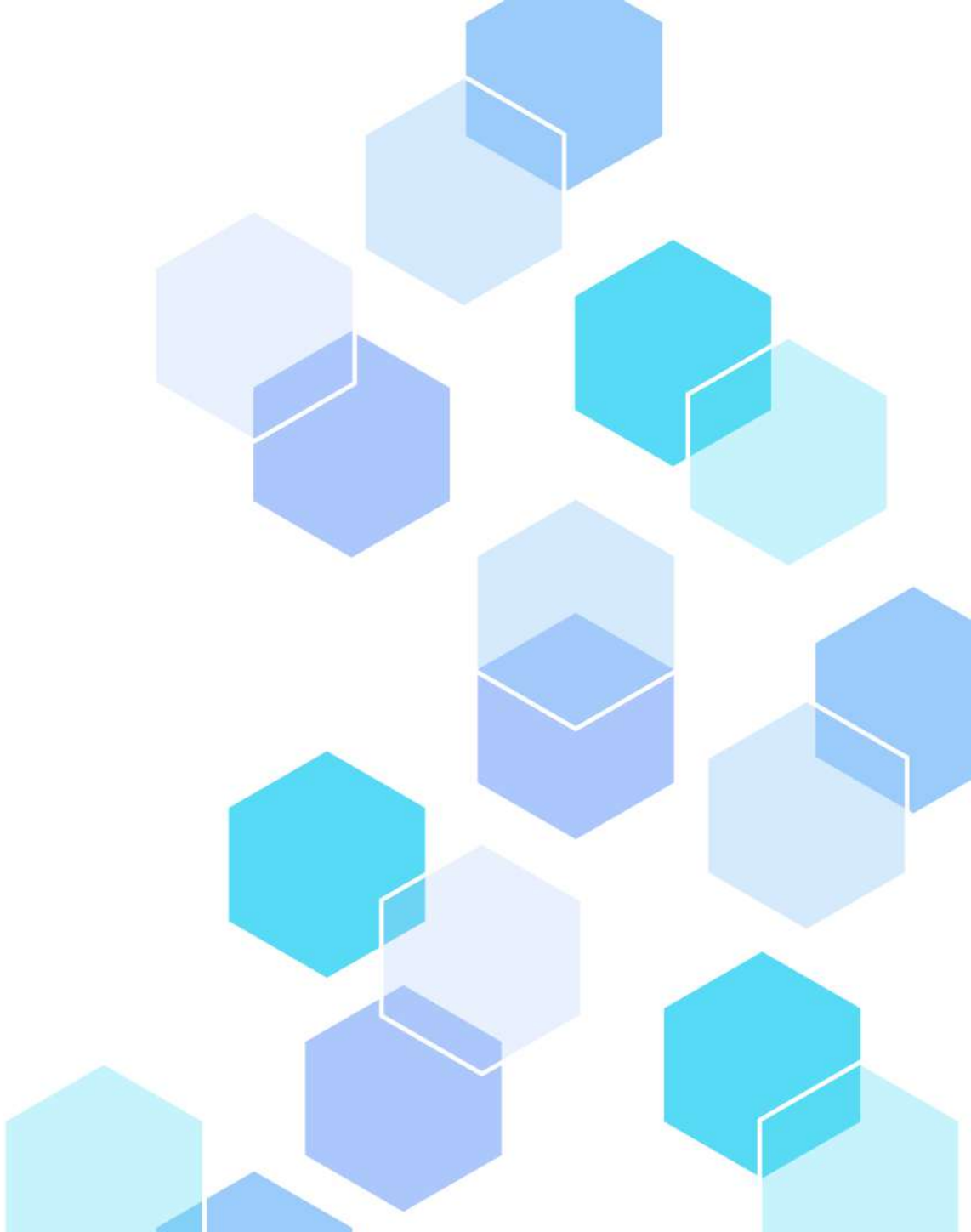
Basic Program Structure of C++

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

int main()
{
  cout << "Hello World";

  return 0;
}
```

02

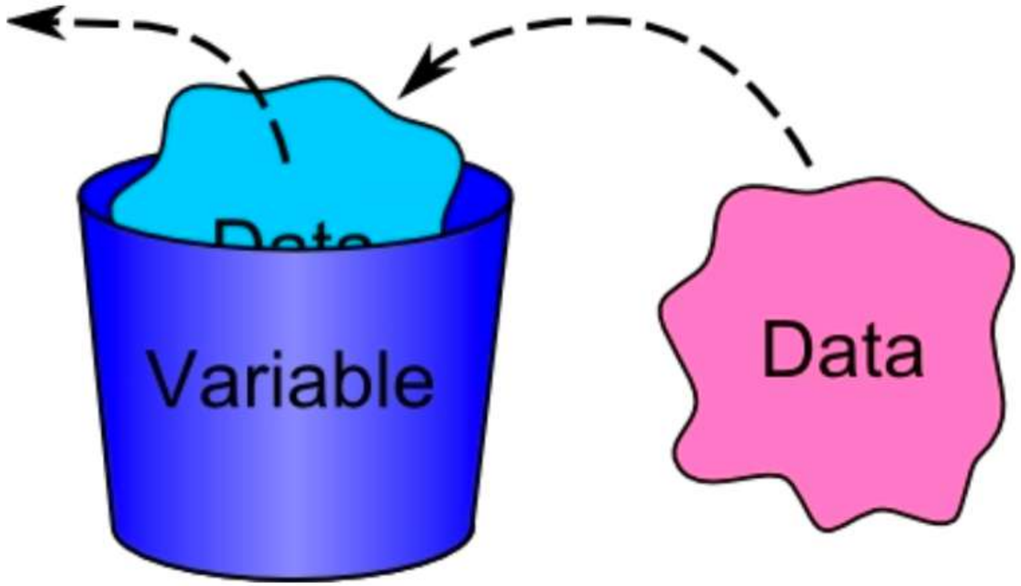
Variables

Data Types, Variables, Keywords, and Constant



Variable

A **container** which **stores some data**.



Examples of Variables

Fruit & Veg. Containers

Glass of Water

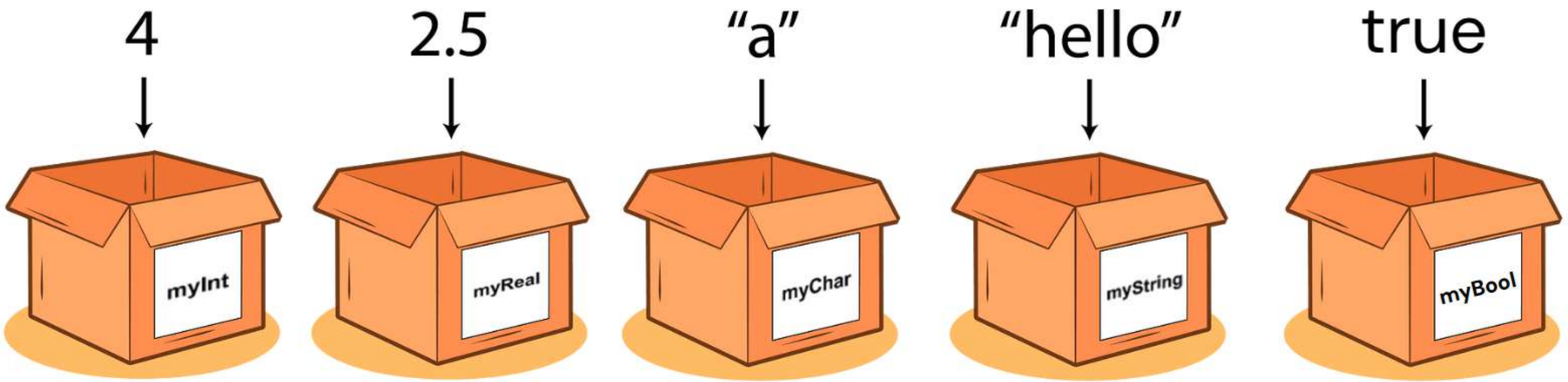
Duster

Data Type

Data types are the **type of data stored** in a C++ program.

Data types are **used** while **defining a variable** or **functions** in C++.

Examples of Data Types



Integer
(**int v = 4;**)

float
(**float v = 2.5;**)

character
(**char v = ‘a’;**)

string
(**char[] v = “hello”;**)
(**string v = “hello”;**)

boolean
(**bool v = true;**)

Keyword

A **reserved word** which have a **specific meaning**.

auto	bool	break	case	catch	char	class
const	continue	double	default	delete	else	enum
explicit	friend	float	for	int	long	mutable
new	operator	private	protected	public	register	return
struct	switch	short	sizeof	static	this	typedef
throw	true	try	union	virtual	void	while

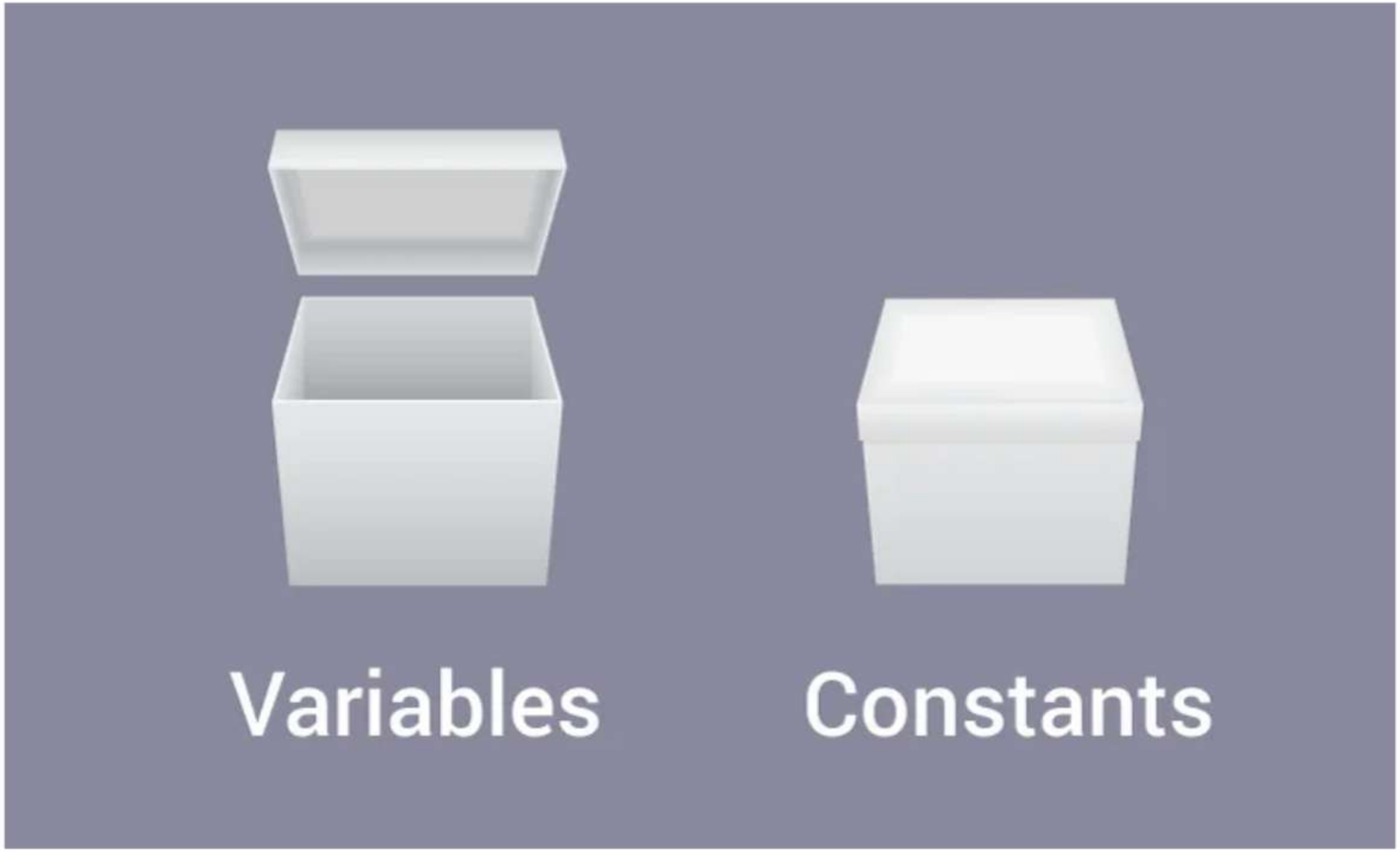
Constant

A constant is a kind of **variable** which **value is fixed** after initialization.

Constant always keeps value unchanged.

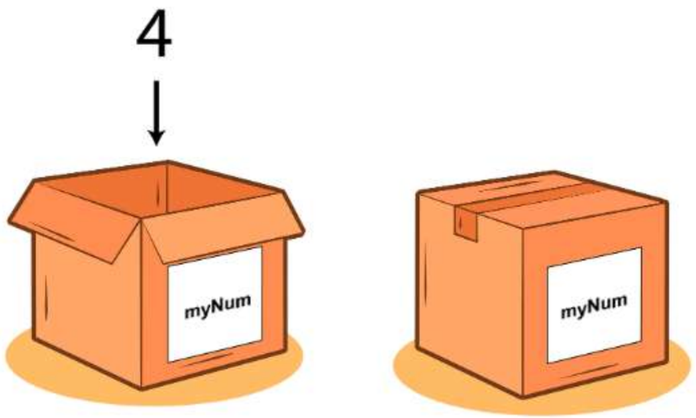
Constant value **can’t be changed**.

Constant



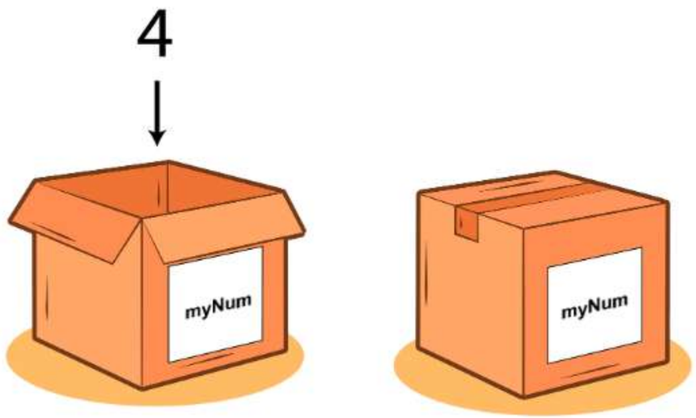
How to create a Constant?

A constant can be created using two ways:




```
const int myNum = 4;
myNum = 10;  // error, can't be changed
```

Using **const** keyword




```
#define myNum 4
myNum = 10;  // error, can't be changed
```

Using a **macro**

Rules of Constant

Constant **value** must have to **initialized while declaration** of variable.

```
const int myNum = 4;
      or
#define myNum 4
```

```
const int myNum;
   myNum = 4;
      or
#define myNum
   myNum = 4
```

Rules of Constant

Constant variable name should have **capitalized** as per global convention.

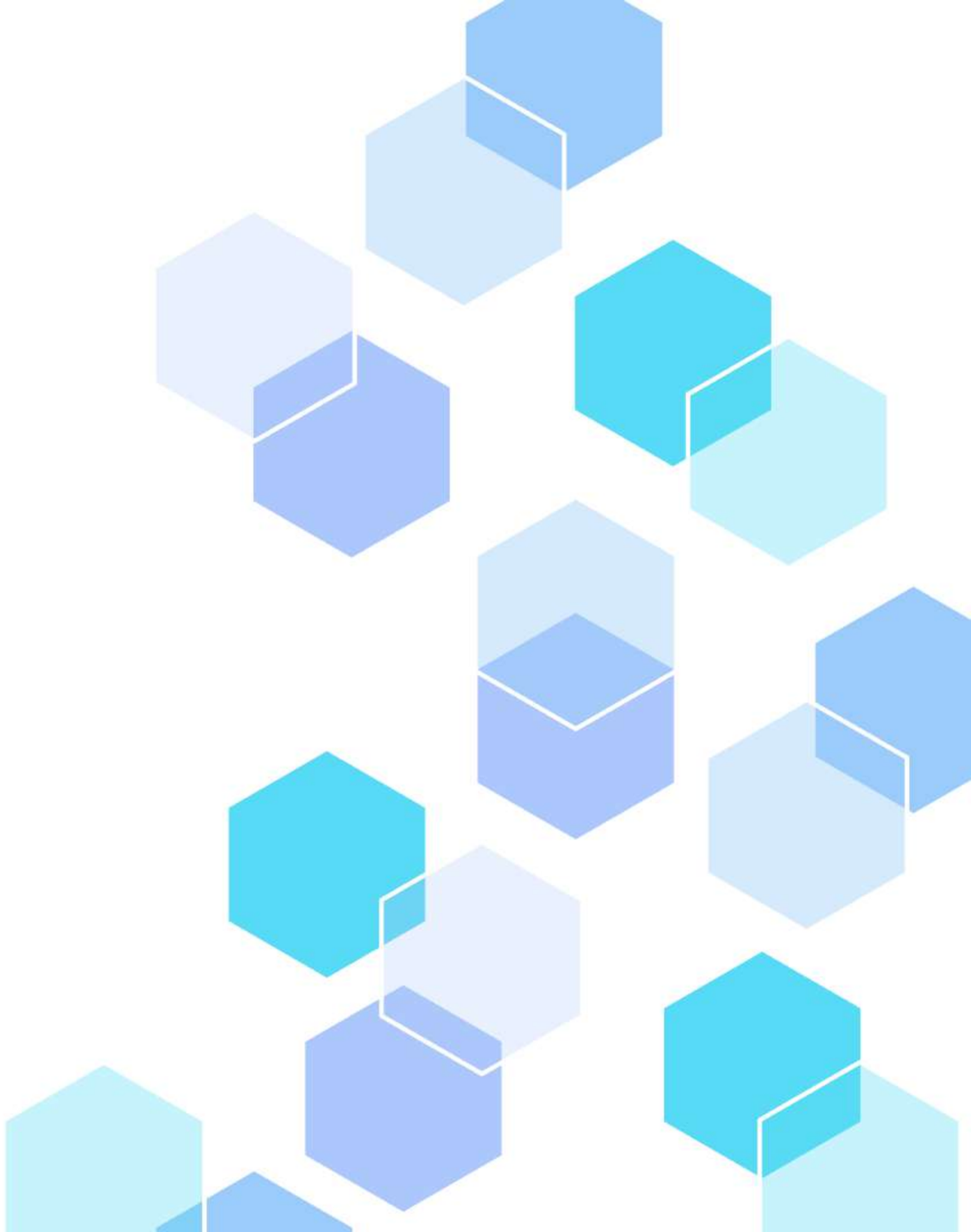
```
const int MAX = 999;
or
#define MAX 999
```

```
const int max = 999;
or
#define max 999
```

03

Objects

cout & cin objects,
endl manipulator



cout object

The cout object in C++ is an **object of class iostream** which is defined in **iostream header file**.

It is used to **display the output to the console**.

The data needed to be displayed on the screen is inserted in the standard output stream (**cout**) using the **insertion operator**(**<<**).

```
cout << msg;

cout << variable;
```

cin object

The cin object in C++ is an **object of class iostream** which is defined in **iostream header file**.

It is used to a**ccept the input** from the standard input device i.e. **keyboard**.

The **extraction operator**(**>>**) is used along with the object cin for **reading inputs**.

```
cin >> variable;
```

endl manipulator

The endl stands for **end of line**.

The use of the endl manipulator takes place to **move the cursor to the beginning of the next line**.

Moreover, its working is similar to the **'\n' escape sequence character**.

```
cin >> msg >> endl;
```

TL;DR

Program Structure

- Library/Header File
- Namespace declaration
- Block of Code

Variables

- Data Types & Variables
- Keywords
- Constants

TL;DR