Installation and Setup

C++

C++

- C++ is a high-level, general-purpose programming language that was developed by Bjarne Stroustrup in 1983 as an extension of the C language.

Features

- Combines object-oriented and procedural programming paradigms.
- Provides low-level memory manipulation capabilities.

Uses

- System/application software.
- Games.
- Embedded systems.

Why Learn C++?

Powerful and Efficient Programming Language

Widely Used in Industries Like Gaming, Finance, and Operating Systems Object-Oriented Features for Code Organization and Reusability

Object-Oriented Features for Code Organization and Reusability

Low-Level Memory Manipulation Capabilities

A Stepping Stone for Learning Other Programming Languages

Requirements

- Compiler – GCC Mingw
- Text Editor – VS Code

Basic Program Structure

```
// Documentation Section

// Link Section

// Definition Section

// Global Declaration Section

// Class Definition

// Main Class Definition

return_type main ()

{

        // statements

}
```

Documentation Section

- Name of the program
- The Author
- Algorithms
- Methods used and
- Other details

Link Section

- Instruction to compiler to link classes, function and operators with program from the system library
- E.g.
    - #include <iostream>

Definition Section

- Symbolic Constants are defined in this section
- E.g.
  - #define PI 3.14

Global Declaration Function

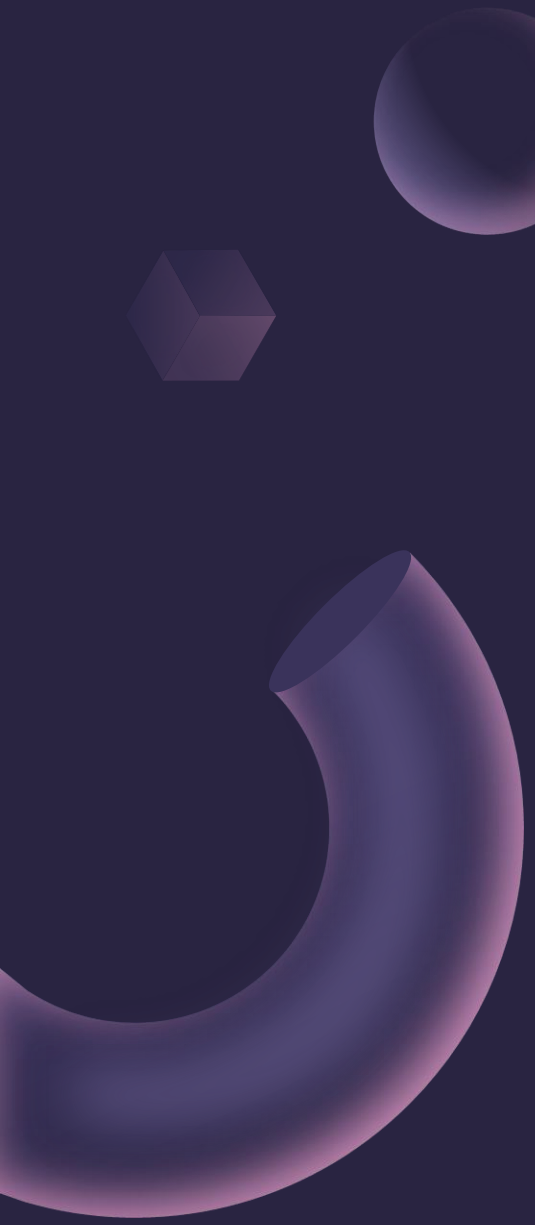
- The variables which is used in more than one functions or blocks are called global variables.

First Program

```
// A C++ Program to display “Hello World!”

// Program coded by Rishikesh Paudel

#include <iostream>

using namespace std;

int main() {

        cout << “Hello World!”;

        return 0;

}
```

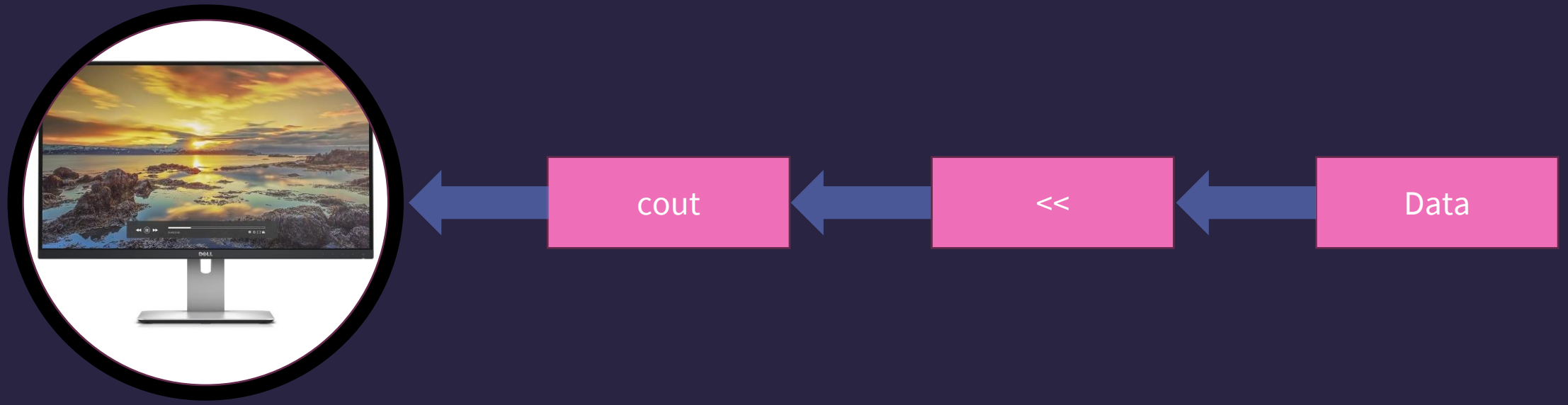
Comments in C++

- Single Line Comments:
  - //
- Multiple Line Comments:
  - /* Your Comment goes here */

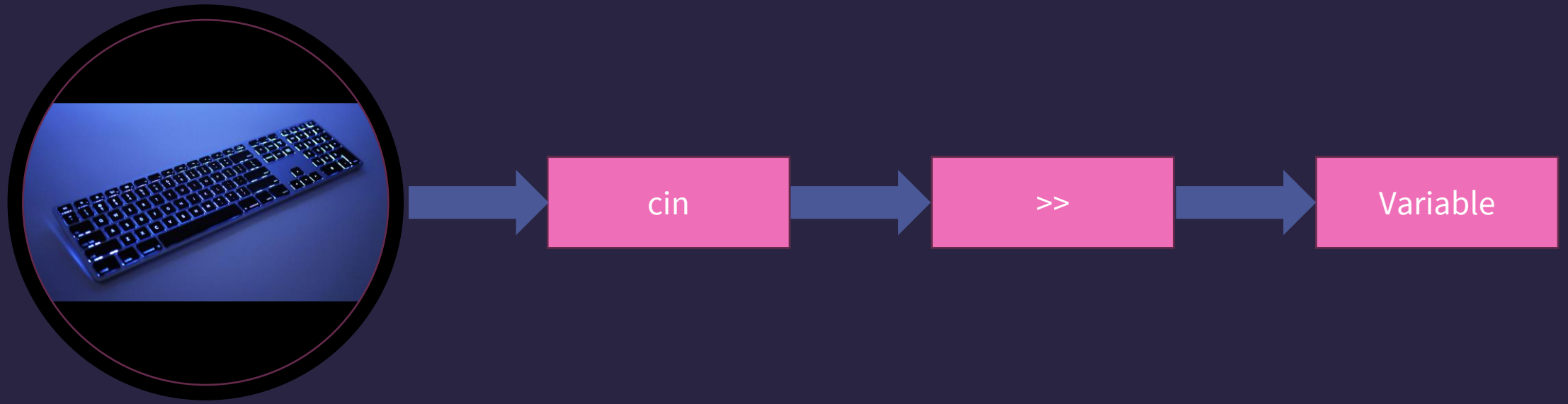


Console Output and Input

Insertion (put) Operator



Extraction (get) Operator



Variables and Data Types

- Storage place in the memory

Name	Description	Memory Size	Range
char	Character	1 Byte	0 to 255
short int	Short Integer	2 Bytes	Signed: -32768 to 32767 Unsigned: 0 to 65535
long int	Long Integer	4 Bytes	Signed: -2147483648 to 2147483647 Unsigned: 0 to 4294967395
float	Floating Point number	4 Bytes	+/- 3.4e+/-38
double	Double precision floating point number	8 Bytes	+/-1.7e+/-308
long double	Long double precision floating point number	10 Bytes	$3.4x10^{-4932}$ to $1.1x10^{+4932}$

The values shown in the above table are for 32 bit computer system

Calculations

- 1 bit: 0 or 1
- 1 Byte : 8 bits => (0000 0000 to 1111 1111) => (0 to 255)
- 2 Bytes : 16 bits => (0000 0000 0000 0000 to 1111 1111 1111 1111) => (0 to 65535)
- 4 Bytes: 32 bits
- => (0000 0000 0000 0000 0000 0000 0000 0000 to 1111 1111 1111 1111 1111 1111 1111 1111)
- => (0 to 4294967295)

Rules for variable declaration

1. The variable name should start with only letters and underscores.
   - int a; // valid
   - int a1; // valid
   - int 1a; // invalid
   - float centigrade_temperature //valid
   - int _a; // valid

Rules for variable declaration

2. The variable name should not be a keyword.
    - int if; // invalid
    - float main; // invalid

Rules for variable declaration

3. White spaces are not allowed between characters of variable, but underscores are allowed.
   - float age of student; // invalid
   - float age_of_student; // valid

Rules for variable declaration

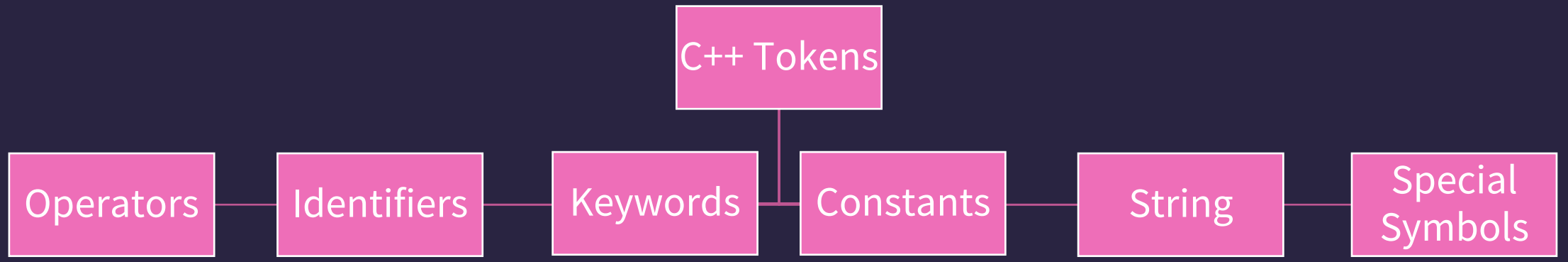
4. The variable name is case sensitive (i.e. uppercase and lowercase are different).
    - The variable temp is not same as Temp or TEMP.

Rules for variable declaration

5. No two variables of the same name are allowed to be declared in the same scope.

Tokens

- Basic elements recognized by C++ compiler



Keywords

- Predefined words in C++
- E.g. int, long, char, do, goto, void, … , etc.

Identifiers

- Variables, Functions, Classes, Arrays, Pointers and symbolic constants

Constants

- Fixed values that do not change during the execution of a program
- Two Types:
  - Defined Constant
  - Declared Constant

Defined Constant

- Using define keyword
- E.g. #define PI 3.14

Declared Constant

- Using const prefix
- E.g. const pi = 3.14

Recap

- Requirements and Setup
- Basic Structure
- Comments
- Console Input and Output
- Variables and data types
- Rules for naming variables
- Tokens
- Keywords
- Identifiers
- Constants
  - Defined
  - Declared

Next Class

- Array
- Pointers
- Dynamic Memory
- Manipulators
- Enumeration
- String
- Reference Variable