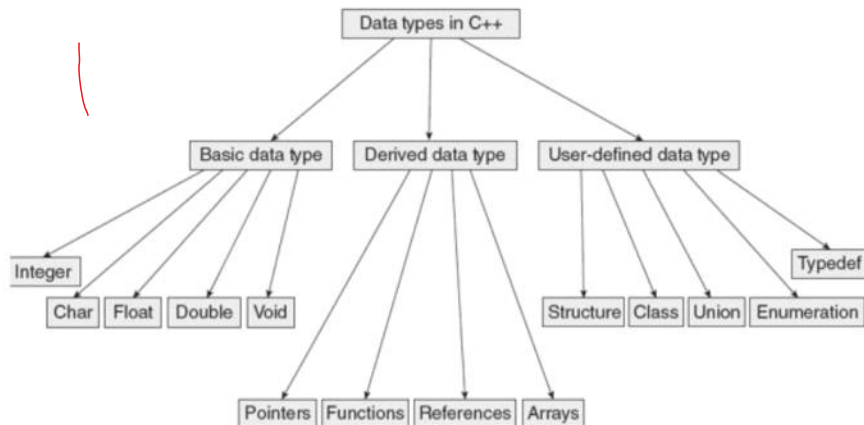


3. Basic Data Types



3. Basic Data Types

Data Type	Size in Bytes	Range
char	1	-128 to 127
unsigned char	1	0 to 255
signed char	1	-128 to 127
int	2	-32768 to 32767
unsigned int	2	0 to 65535
signed int	2	-32768 to 32767
short int	2	-32768 to 32767
unsigned short int	2	0 to 65535
signed short int	2	-32768 to 32767
long int	4	-2147483648 to 2147483647
unsigned long int	4	0 to 4294967295
signed long int	4	-2147483648 to 2147483647
float	4	3.4E-38 to 3.4E+38
double	8	1.7E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

4. Variables & Constant

4. Variables & Constant

- Variable is defined as a meaningful name given to the data storage location in computer memory.
 - When using a variable, we actually refer to address of the memory where the data is stored.
 - Numeric variables can be used to store either integer values or floating point values.
 - Character variables can include any letter from the alphabet or from the ASCII chart and numbers 0 – 9 inserted between single quotes.
 - In C++, a number that is put in single quotes is not the same as a number without them.
-
- Each variable to be used in the program must be declared. To declare a variable, specify the data type of the variable, followed by its name.
 - In C++, variables are declared at three basic places as follows:
 - First, when a variable is declared inside a function it is known as a local variable.
 - Second, when a variable is declared in the definition of function parameters, it is known as formal parameter (we will study this in Chapter 4).
 - Third, when the variable is declared outside all functions, it is known as a global variable.

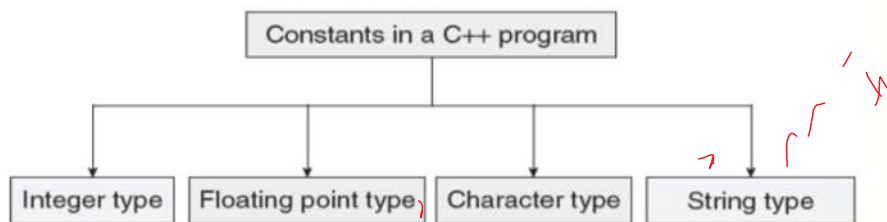
data_type variable_name;

int emp_num = 7;
✓ float salary = 5000;
char grade = 'A';
double balance_amount = 100000000;

int emp_num;
float salary;
char grade;
double balance_amount;
unsigned short int acc_no;

4. Variables & Constant

- **Constants are identifiers whose value does not change. While variables can change their value at any time, constants can never change their value.**
- **Constants are used to define fixed values such as Pi or the charge on an electron so that their value does not get changed in the program even by mistake.**
- **A constant is an explicit data value specified by the programmer.**
- **The value of the constant is known to the compiler at the compile time.**



4. Variables & Constant

- **Rule 1 Constant names are usually written in capital letters to visually distinguish them from other variable names which are normally written in lower case characters.**
- **Rule 2 No blank spaces are permitted in between the # symbol and define keyword.**
- **Rule 3 Blank space must be used between #define and constant name and between constant name and constant value.**
- **Rule 4 #define is a preprocessor compiler directive and not a statement. Therefore, it does not end with a semicolon.**

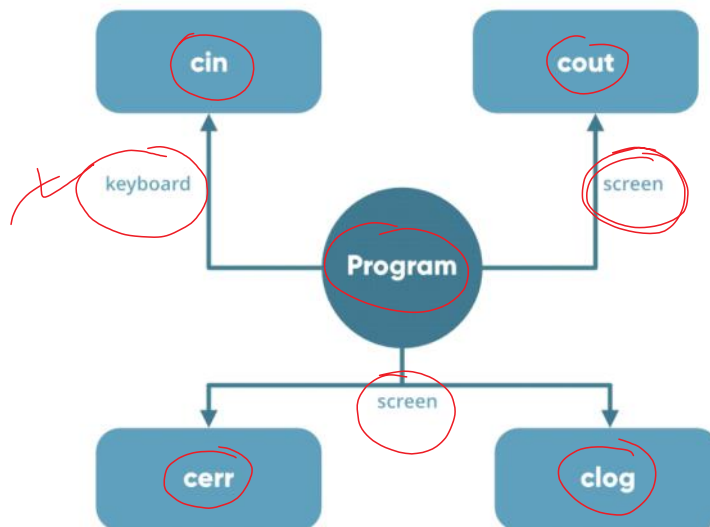
`const data_type const_name = value;`

The const keyword specifies that the value of pi cannot change.

`const float pi = 3.14;`

`#define PI 3.14159`
`#define service_tax 0.12`

6. Input-Output in C++



6. Input-Output in C++

- ✓ **Input Stream:** If the direction of flow of bytes is from the device (for example, Keyboard) to the main memory then this process is called input.
- ✓ **Output Stream:** If the direction of flow of bytes is opposite, i.e. from main memory to device (display screen) then this process is called output.

6. Input-Output Header

- **iostream**: iostream stands for standard input-output stream. This header file contains definitions of objects like cin, cout, cerr, etc.
- **iomanip**: iomanip stands for input-output manipulators. The methods declared in these files are used for manipulating streams. This file contains definitions of setw, setprecision, etc.
- **fstream**: This header file mainly describes the file stream. This header file is used to handle the data being read from a file as input or data being written into the file as output.
- **bits/stdc++**: This header file includes every standard library. In programming contests, using this file is a good idea, when you want to reduce the time wasted in doing chores; especially when your rank is time sensitive. To know more about this header file refer [this](#) article.

6. Input-Output Stream

Standard output stream (cout): Usually the standard output device is the display screen. The C++ cout statement is the instance of the ostream class. It is used to produce output on the standard output device which is usually the display screen. The data needed to be displayed on the screen is inserted in the standard output stream (cout) using the insertion operator(<<).

cout <<

Standard input stream (cin): Usually the input device in a computer is the keyboard. C++ cin statement is the instance of the class **istream** and is used to read input from the standard input device which is usually a keyboard. The **extraction operator(>>)** is used along with the object **cin** for reading inputs. The extraction operator extracts the data from the object **cin** which is entered using the keyboard

cin >>

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      char sample[] = "Object Oriented Programming";
8
9      cout << sample << endl;
10
11     return 0;
12 }
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age;
7
8      cout << "Enter your age:";
9      cin >> age;
10     cout << "\nYour age is: " << age;
11
12     return 0;
13 }
```