

Console Input/Output

fit@hcmus | Programming 1 | 2023



Console Input/Output

- o Using these objects: std::cin, std::cout, std::cerr of iostream
- Declaring before use:

```
#include <iostream>
```

fit@hcmus | Programming 1 | 2023



Input Using std::cin



- o std::cin is used with >> to gather input
- O The stream extraction operator is >>
- O Using more than one variable in std::cin allows more than one value to be read at a time
- Examples:
 - std::cin >> miles;
 - std::cin >> numberofLanguages;
 - std::cin >> dragrons >> trolls;
 - std::cin >> dragrons
 - >> trolls;

fit@hcmus | Programming 1 | 2023





Input Using std::cin

- o std::in stops when getting white spaces.
- o Try to use function getline of std::cin.

fit@hcmus | Programming 1 | 2023

6



Input Using std::cin

```
#include <iostream>
#include <cstring>
int main() {
    char name[80];
    std::cout << "Input your name: ";
    std::cin.getline(name, 80);
    std::cout << "Your name is " << name << "\n";
    return 0;
}</pre>
```

fit@hcmus | Programming 1 | 2023



Output Using std::cout

- Any combinations of variables and strings can be output.
- o std::cout is used with << to output.
- The stream insertion operator is <<
- Expression evaluated and its value is printed at the current cursor position on the screen.

fit@hcmus | Programming 1 | 2023



8





- The new line character is '\n'. May appear anywhere in the string.
- std::endl causes insertion point to move to beginning of next line.

fit@hcmus | Programming 1 | 2023



Output Using std::cout

Commonly used escape sequences:

	Escape Sequence	Description
\n	Newline	Cursor moves to the beginning of the next line
\t	Tab	Cursor moves to the next tab stop
\b	Backspace	Cursor moves one space to the left
\r	Return	Cursor moves to the beginning of the current line (not the next line)
\\	Backslash	Backslash is printed
\'	Single quotation	Single quotation mark is printed
\ "	Double quotation	Double quotation mark is printed

fit@hcmus | Programming 1 | 2023



10

Output Using std::cout



o Formatting for Numbers:

```
std::cout.setf(std::ios::fixed);
std::cout.setf(std::ios::showpoint);
std::cout.precision(2);
std::cout.setf(std::ios::fixed);
std::cout.setf(std::ios::showpoint);
std::cout.precision(2);
std::cout << 7.9999 << " " << 10.5 << std::endl;</pre>
```

8.00 10.50

fit@hcmus | Programming 1 | 2023

0-0-011



Output Using std::cout

```
#include <iostream>
   #include <iomanip>
   int main()
3
4 ₹ {
        double pi_v = 3.14159, npi = -3.14159;
6
        std::cout << std::fixed << std::setprecision(0) << pi v << " " << npi << std::endl;</pre>
7
        std::cout << std::fixed << std::setprecision(1) << pi_v << " " << npi<< std::endl;
8
        std::cout << std::fixed << std::setprecision(3) << pi_v << " " << npi << std::endl;
9
        std::cout << std::fixed << std::setprecision(4) << pi_v << " " << npi << std::endl;
10
        std::cout << std::fixed << std::setprecision(5) << pi_v << " " << npi << std::endl;
11
        std::cout << std::scientific << std::setprecision(6) << pi_v << " " << npi << std::endl;
13
                                              3 -3
        return 0;
14
                                              3.1 -3.1
                                              3.142 -3.142
                                              3.1416 -3.1416
                                              3.14159 -3.14159
                                              3.141590e+00 -3.141590e+00
    fit@hcmus | Programming 1 | 2023
```

12

File

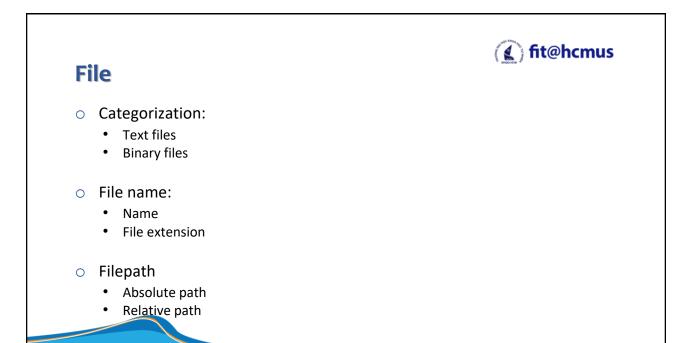


- A file is a container in a computer system for **storing information**.
- There are different types of files such as text files, data files, directory files, binary and graphic files, etc.
- Files can be stored on optical drives, hard drives or other types of storage devices.

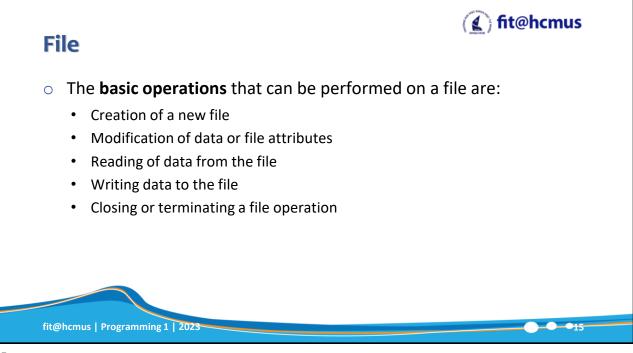
(techopedia)

fit@hcmus | Programming 1 | 2023





fit@hcmus | Programming 1 | 2023





Reading from a Text File

- o using std::ifstream.
 - Open file for reading: open
 - Close file after reading: close
 - Take input (same as cin, extractor operator): >>
- o including fstream

fit@hcmus | Programming 1 | 2023

16

16

Writing to a Text File



- o using std::ofstream.
 - Open file for writing: open
 - Close file after writing: close
 - Take input (same as cout, insertion operator): <<
- \circ including fstream

fit@hcmus | Programming 1 | 2023

```
( fit@hcmus
                        1 #include <iostream>
                        2 #include <fstream>
Examples
                        3
4 int main()
                        5 ₩ {
                                  std::ifstream fIn;
                                  fIn.open("Data01.txt");
                                  if (fIn.is_open() == false)
                        10 v
                                       std::cout << "File does not exist" << std::endl;
                        12
                                       return 1;
                        13 A
                                 }
                        14
                        15
                                  int N, i;
                        16
                                  int A[100];
                        18
                                  fIn >> N;
                                  for (i = 0; i < N; i++)
                        19
                                       fIn >> A[i];
                        20
                        21
                        22
                                  for (i = 0; i < N; i++)
                        23
                                       std::cout << A[i] << "\t";
                        24
                                  std::cout << "\n";
                        25
                                  fIn.close();
                        26
                                  std::cout << "Done\n";
                        27
                        28
                                  return 0;
fit@hcmus | Programming 1 | 2023
```

18

```
( fit@hcmus
                                 #include <fstream>
                                 #include <iostream>
Examples
                                 int main () {
                                    char data[100];
                                    // open a file in write mode.
                             8
                                    std::ofstream outfile;
                                    outfile.open("afile.txt");
                            10
                                    std::cout << "Writing to the file" << std::endl;</pre>
                                    std::cout << "Enter your name: ";
                            13
                                    std::cin.getline(data, 100);
                            14
                                    // write inputted data into the file.
                            16
                                    outfile << data << std::endl;
                            18
                                    int age;
                            19
                            20
                                    std::cout << "Enter your age: ";</pre>
                            21
                                    std::cin >> age;
                                    // again write inputted data into the file.
                            24
                                    outfile << age << std::endl;</pre>
                            25
                            26
                                    // close the opened file.
                            27
                                    outfile.close();
fit@hcmus | Programming 1 | 2023
```

