C++

TIN ĐẠI CƯƠNG

BÀI 14: CÁC KIẾN THỨC BỔ SUNG

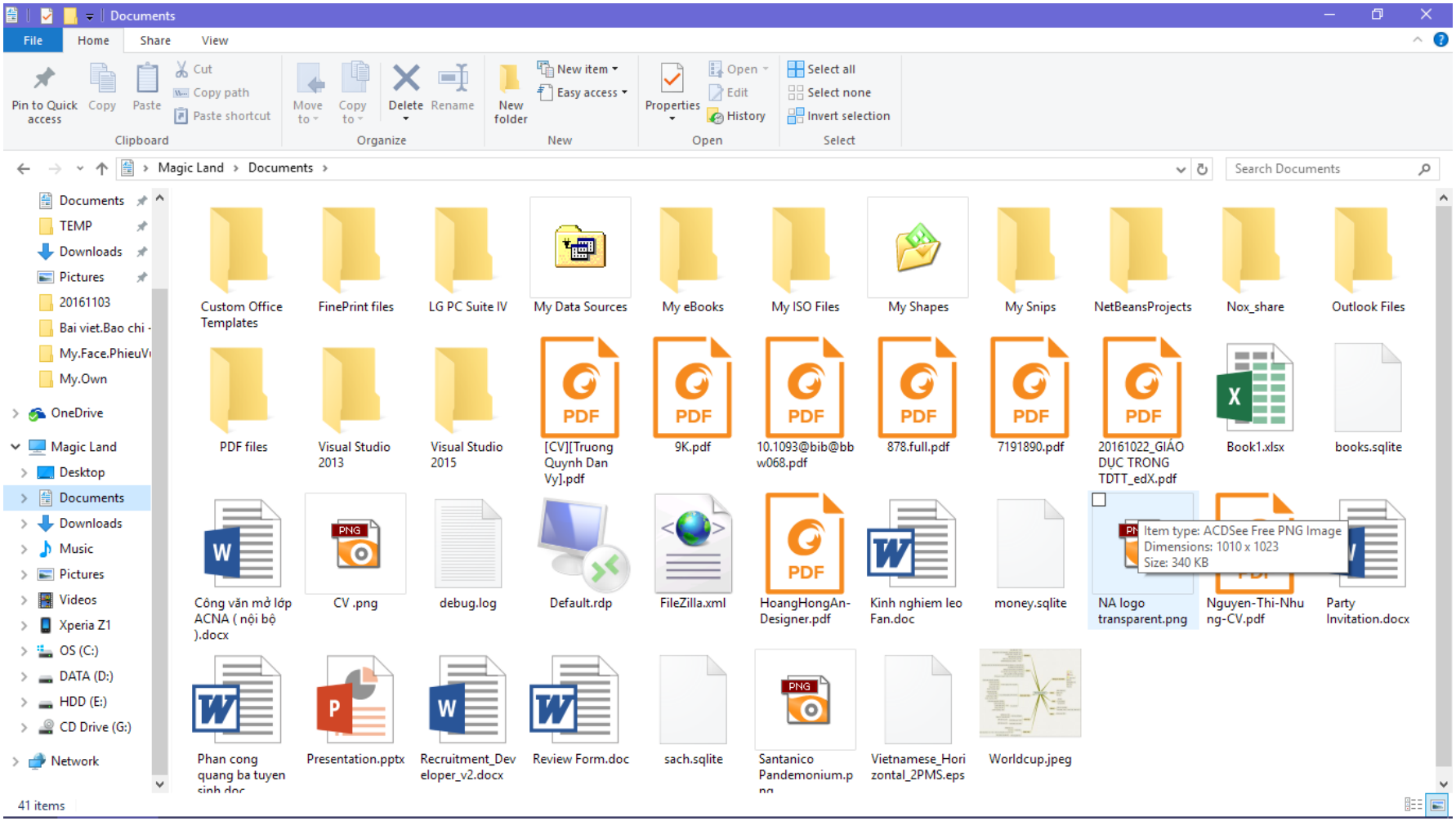
Nội dung

1. Làm việc với tập tin văn bản
2. Mảng nhiều chiều
3. Các hàm toán học và lượng giác
4. Các hàm xử lý kí tự

Phần 1

Làm việc với tập tin văn bản

Làm việc với tập tin



Làm việc với tập tin văn bản

- Tập tin, còn gọi là file, là thành phần cơ bản của các thiết bị lưu trữ
- C++ chia tập tin làm 2 loại:
  - Tập tin dạng nhị phân (binary file): có thể xem như dãy các byte, đọc/ghi theo từng byte
  - Tập tin dạng văn bản (text file): có thể xem như dãy các string, đọc/ghi theo từng dòng
- Các biến cin, cout thực chất là các tập tin văn bản đặc biệt
  - Làm việc với file văn bản cũng tương tự làm việc với cin, cout

Ghi chuỗi ra tập tin văn bản

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    // khai báo biến có kiểu tập tin văn bản để ghi ra
    ofstream myfile;
    // mở tập tin có tên là "example.txt"
    myfile.open("example.txt");
    // ghi 100 dòng vào tập tin
    for (int i = 0; i < 100; i++)
        myfile << "Dong thu " << i << endl;
    // đóng tập tin lại
    myfile.close();
}
```

Đọc chuỗi từ tập tin văn bản

```
int main() {
    string line;
    // khai báo biến có kiểu tập tin văn bản để đọc vào
    ifstream myfile;
    // mở tập tin có tên là "example.txt"
    myfile.open("example.txt");
    // đọc hết các dòng của tập tin và in ra
    while (!myfile.eof()) {
        getline(myfile, line)
        cout << line << endl;
    }
    // đóng tập tin lại
    myfile.close();
}
```

Phần 2

Mảng nhiều chiều

Mảng nhiều chiều

- Mảng gốc trong C/C++ cho phép khai báo dữ liệu có nhiều chiều (2,3,4 hoặc hơn) là dạng mở rộng của kiểu mảng
- Nếu sử dụng vector, có thể xem mảng nhiều chiều là các vector lồng nhau
- Mảng gốc 2 chiều: int a[3][4] // 3 dòng x 4 cột
- Vector: vector<vector<int>> a;

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Mảng gốc nhiều chiều

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

int main () {
    // mảng 2 chiều: 5 hàng x 2 cột
    int a[5][2] = { {0,0}, {1,2}, {2,4}, {3,6}, {4,8} };
    // duyệt theo từng hàng
    for (int i = 0; i < 5; i++) {
        // mỗi hàng lại duyệt theo từng cột
        for (int j = 0; j < 2; j++) cout << a[i][j] << " ";
        // in hết một hàng thì xuống dòng
        cout << endl;
   }
}
```

Phần 3

Các hàm toán học và lượng giác

Thư viện cmath

- Hầu hết các hàm toán học và lượng giác được khai báo trong thư viện cmath
- Tham khảo đầy đủ trong phụ lục B.1
- Tất cả các hàm lượng giác đều làm việc với đơn vị đo là radian
- Rất thận trọng khi xử lý dữ liệu số, đặc biệt là số thực, một số tình huống có thể gây nhầm lẫn giữa lập trình viên và trình biên dịch

Thư viện cmath

Hàm	Mục đích sử dụng
abs(x) / fabs(x)	Trả về trị tuyệt đối của x
ceil(x)	Làm tròn lên (số nguyên nhỏ nhất >=x)
floor(x)	Làm tròn xuống (số nguyên lớn nhất <=x)
exp(x)	Trả về e^x
log(x)	Trả về logarit cơ số e của x
log10(x)	Trả về logarit cơ số 10 của x
pow(x, y)	Trả về x^y
sqrt(x)	Trả về x2
fmod(a, b)	Trả về phần dư của phép chia a cho b
sin(x), cos(x), tan(x)	Trả về sin, cos, tang của góc x (đơn vị radian)
asin(x), acos(x), atan(x)	Trả về arcsin, arccos, arctang của x

Phần 4

Các hàm xử lý kí tự

Thư viện cctype

Hàm	Mục đích sử dụng
isalnum(x)	Trả về true nếu x là chữ cái hoặc chữ số
isalpha(x)	Trả về true nếu x là chữ cái (tiếng Anh)
isdigit(x)	Trả về true nếu x là chữ số (từ 0 đến 9)
issuper(x)	Trả về true nếu x là chữ cái viết hoa
tolower(x)	Trả về chữ cái x ở dạng viết thường
toupper(x)	Trả về chữ cái x ở dạng viết hoa