

# TIN HỌC ĐẠI CƯƠNG

---

Bài 1: Giới thiệu môn học & Làm quen  
với Dev-C++

# Nội dung chính

---

1. Giới thiệu môn học
2. Viết chương trình cho máy tính
3. Làm quen với Dev-C++
  1. Các bước viết chương trình
  2. Ngôn ngữ lập trình C++
  3. Công cụ Dev-C++
4. Bài tập

Phần 1

# Giới thiệu môn học

# Giáo trình & Giờ học

- Thời lượng: 3 tín chỉ (15 buổi x 3 tiết)
- Giáo trình chính
  - *“Introduction to Engineering Programming: Solving Problems with Algorithms”* (James Paul Holloway)
  - Đã có bản dịch tiếng Việt
- Công cụ trên máy tính: **Dev-C++ 5.11**
  - Hoặc những công cụ tương đương
- Phần lý thuyết: lý thuyết trên slide + chữa bài tập
- Phần thực hành: viết chương trình trên máy tính

# Nội dung giảng dạy

---

- Khái niệm cơ bản của lập trình C/C++
- Hàm/chương trình con
- Các lệnh cơ bản
- Câu lệnh lặp
- Câu lệnh lựa chọn
- Chuỗi (string)
- Mảng (vector)
- Tập tin (file)
- Bài tập tổng hợp

# Mục tiêu của môn học

---

- Hiểu biết cơ bản về ngôn ngữ lập trình C/C++
- Biết cách triển khai (lập trình) một số thuật toán trên máy tính
- Biết cách viết, dịch, sửa lỗi và chạy một chương trình viết bằng C++
- Biết cách giải một số bài toán đơn giản bằng lập trình C++
- Biết ứng dụng kiến thức lập trình vào những công việc sau này

# Tại sao phải học môn này?

- Hiểu biết hơn về máy tính và lập trình máy tính
- Làm quen với máy tính theo cách của giới làm kỹ thuật
- Hiểu cách thức giải quyết một vấn đề bằng máy tính
- Nâng cao tư duy logic và tư duy thuật toán
- Lấy kiến thức nền cho các môn học tiếp sau của ngành CNTT (\*)
- Lấy bằng đại học

# Thi & Tính điểm

- Tính điểm:
  - Điểm quá trình (50%):
    - Điểm chuyên cần
    - Điểm kiểm tra giữa kỳ (5 đầu điểm)
  - Điểm kiểm tra cuối kì (50%, thi thực hành, máy chấm tự động)
- Như vậy tất cả các điểm số của môn này đều là điểm thực hành trên máy, chú trọng vào viết chương trình, không có lý thuyết học thuộc
- Giảng viên:
  - Họ tên: Trương Xuân Nam, khoa CNTT
  - Email: [namtx@wru.vn](mailto:namtx@wru.vn)  
[truongxuannam@gmail.com](mailto:truongxuannam@gmail.com)



# Một vài chú ý khác

---

- Cần xem trước giáo trình trước giờ học
- Cần xem lại bài cũ trước khi lên lớp
- Phải làm hết bài tập (được giao trên lớp và trong giờ thực hành)
- Yêu cầu hỗ trợ của giáo viên khi cần thiết
- Mọi thông tin cần thiết về môn học được đưa lên <http://txnam.net> mục **BÀI GIẢNG**
- Cách học hợp lý môn này: trao đổi với thầy giáo, không ghi chép nhiều trong giờ lý thuyết

Phần 2

# Viết chương trình cho máy tính

# Máy tính chỉ hiểu con số

- Mọi thông tin đều có thể chuyển về dạng số:

- Các số → giữ nguyên → số
- Âm thanh → số hóa (tần số) → số
- Hình ảnh → số hóa (ma trận điểm) → số
- ...

➔ Máy tính xử lý các thông tin ở dạng số

➔ Mọi thông tin trong máy tính đều được lưu ở dạng số, cụ thể là số ở dạng nhị phân

➔ Ra lệnh cho máy tính phải viết lệnh ở dạng số

# Các lệnh máy là các dãy số

- Máy tính chỉ hiểu một số lệnh cơ bản:
  - Thao tác bộ nhớ: đọc/ghi số
  - Tính toán: cộng 2 số, trừ 2 số,...
  - So sánh: so sánh 2 số với nhau
  - ...
- Chương trình máy tính = dãy các lệnh máy để chỉ thị từng bước làm việc nhỏ
- Kích thước một chương trình máy tính
  - Loại rất nhỏ  $\approx$  vài nghìn lệnh máy
  - Loại nhỏ  $\approx$  vài chục nghìn lệnh máy
  - Loại vừa  $\approx$  vài trăm nghìn lệnh máy
  - Loại lớn  $\approx$  vài triệu lệnh máy

# Thực hiện một chương trình

- **Bước 1:** người dùng ra lệnh cho máy tính thực hiện một chương trình
- **Bước 2:** máy tính đọc file chương trình trên đĩa và nạp chương trình vào bộ nhớ
- **Bước 3:** hệ thống có một số thao tác chuẩn bị để chương trình sẵn sàng chạy
- **Bước 4:** máy tính đọc từng lệnh trong bộ nhớ và thực hiện từng lệnh một
  - Tốc độ thực hiện lên đến hàng tỉ lệnh/giây
  - Một số hệ thống có thể thực hiện nhiều lệnh cùng lúc

# Máy tính thực hiện từng lệnh một

- Chương trình máy tính được ghi trên đĩa ở dạng file chương trình (.COM, .EXE, .DLL,...)
- Máy tính đọc lệnh máy trong bộ nhớ và thực hiện từng lệnh một

00011000 00010000

Nạp số 16 vào ô nhớ số 8

00011001 00001111

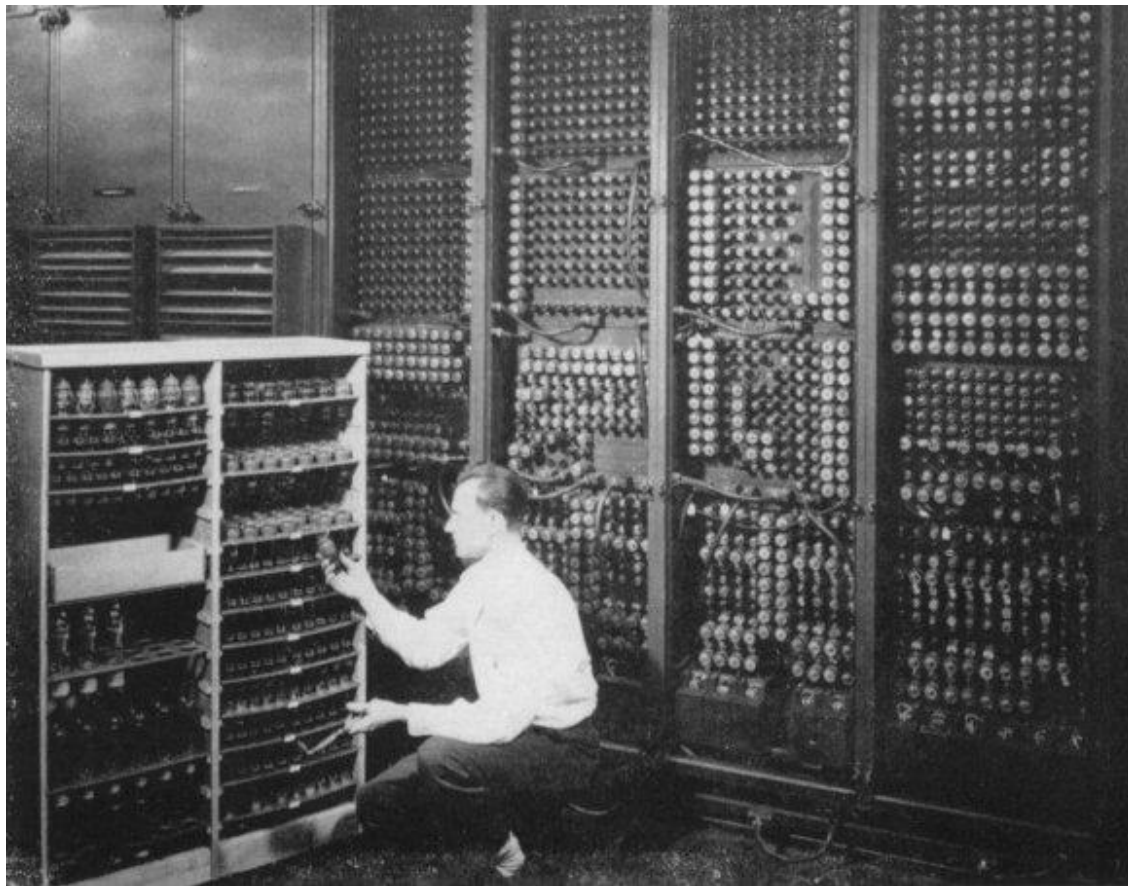
Nạp số 15 vào ô nhớ số 9

00101010 10001001

Cộng hai số ở ô nhớ số 8  
và ô nhớ số 9 sau đó ghi  
kết quả vào ô nhớ số 10

# Viết chương trình = viết dãy số?

- Thời kì đầu: viết trực tiếp lệnh máy (dãy số)
  - Bất lợi: khó hiểu, dễ nhầm lẫn, viết lâu,...



# Viết chương trình = viết dãy số?

- Hợp ngữ: sử dụng các kí hiệu đơn giản bằng tiếng Anh, gần gũi với lệnh máy
  - Bất lợi: người lập trình phải biết rõ về từng lệnh máy, viết dài, dễ nhầm lẫn





# Viết chương trình = viết dãy số?

- Ngôn ngữ lập trình bậc cao: các lệnh ở dạng gần gũi với ngôn ngữ tự nhiên, **trình biên dịch** chuyển một lệnh này thành các lệnh máy
  - Ngôn ngữ bậc cao đơn giản: BASIC, FORTRAN,...
  - Ngôn ngữ lập trình thủ tục: ALGOL, PASCAL, C,...
  - Ngôn ngữ lập trình hướng đối tượng: SmallTalk, C++, Object Pascal, Java, C#,...
- Các ngôn ngữ lập trình đặc biệt (dùng cho những mục đích riêng): Prolog, SQL,...

Phần 3

# Làm quen với Dev-C++

# 3.1 Các bước viết chương trình

- Một chương trình máy tính được xây dựng để giải quyết một bài toán cụ thể nào đó
- Việc xây dựng một chương trình máy tính luôn tuân theo các bước sau:
  - **Bước 1:** xác định (mô tả) bài toán cần giải quyết
  - **Bước 2:** xây dựng lời giải (thuật toán)
  - **Bước 3:** chuyển lời giải bài toán thành chương trình viết bằng một ngôn ngữ lập trình nào đó
  - **Bước 4:** dịch chương trình thành dạng mã máy để máy tính có thể thực hiện được

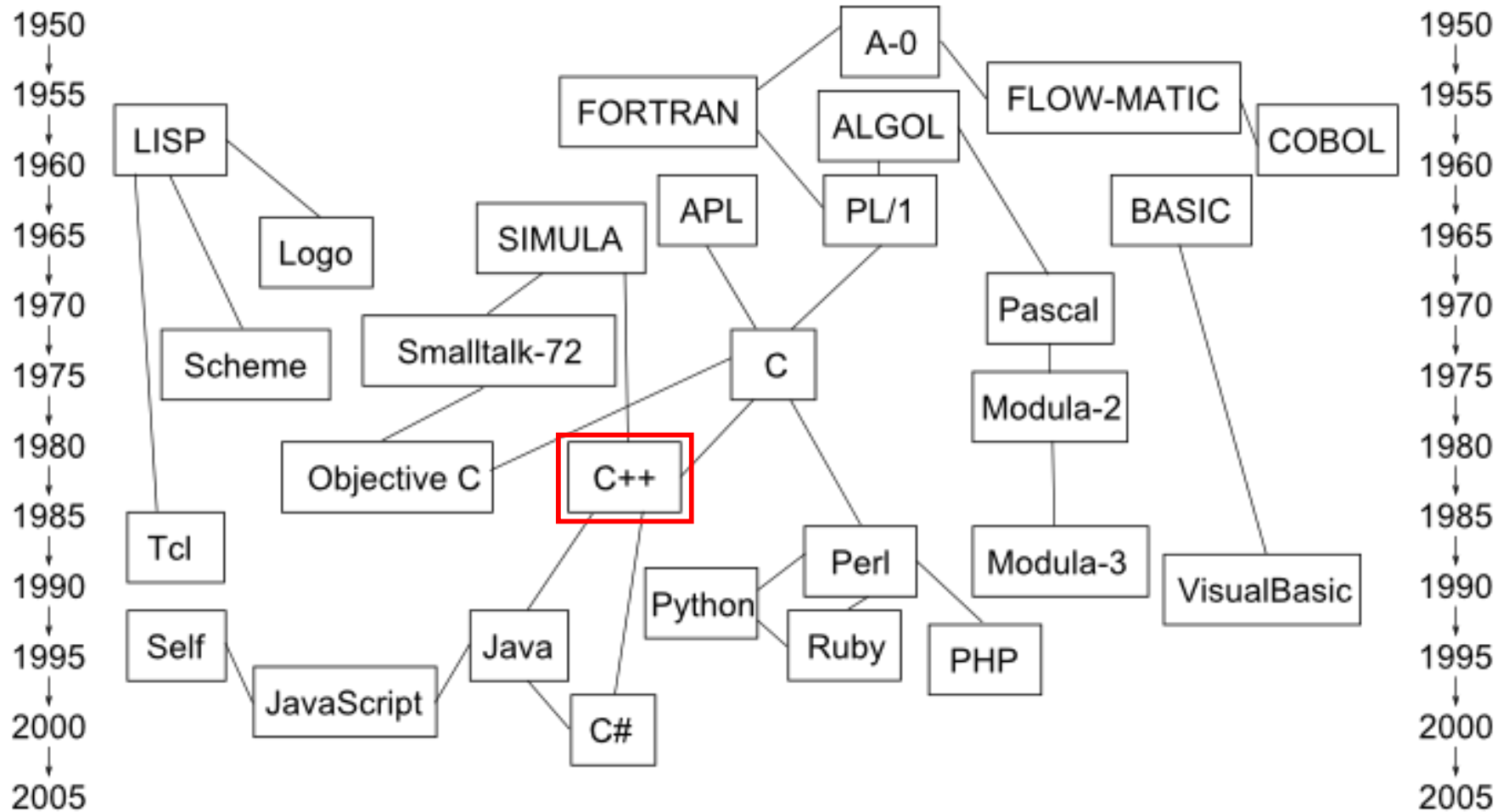
# 3.1 Các bước viết chương trình

- Bước 1 - xác định (mô tả) bài toán cần giải quyết:
  - Ví dụ: bài toán tính  $A^2$
  - Xác định bài toán: người dùng cho số  $A$ , máy tính cần tính  $A^2$  dựa trên số  $A$  đã biết
- Bước 2 - xây dựng lời giải (thuật toán):
  - Có nhiều cách mô tả thuật toán (bằng lời hoặc bằng sơ đồ khối)
  - Ví dụ (mô tả bằng lời): nhập  $A$  từ bàn phím, sau đó tính giá trị  $A \times A$  và in kết quả ra màn hình

# 3.1 Các bước viết chương trình

- Bước 3 - chuyển lời giải bài toán thành chương trình viết bằng một ngôn ngữ lập trình nào đó:
  - Chọn ngôn ngữ lập trình thích hợp với bài toán
  - Viết chương trình theo thuật toán đã định
- Bước 4 - dịch chương trình thành dạng mã máy để máy tính có thể thực hiện được:
  - Sử dụng trình biên dịch của ngôn ngữ đã chọn và dịch chương trình sang dạng mã máy
  - Nếu xảy ra lỗi, tìm và sửa lỗi trong chương trình sau đó dịch lại đến khi không còn lỗi nữa

## 3.2 Ngôn ngữ lập trình C/C++



## 3.2 Ngôn ngữ lập trình C/C++

- Tác giả: Bjarne Stroustrup (Mỹ)
- Ý tưởng bắt đầu từ năm 1979
- Được giới thiệu năm 1985
- Phiên bản C++ 2.0 năm 1989
- Phiên bản mới nhất: C++17
- Môn học này chỉ học khoảng 10% kiến thức về C++ và các thư viện của nó
- Cần 3-5 năm để trở thành lập trình viên C++ ở mức độ chuyên nghiệp



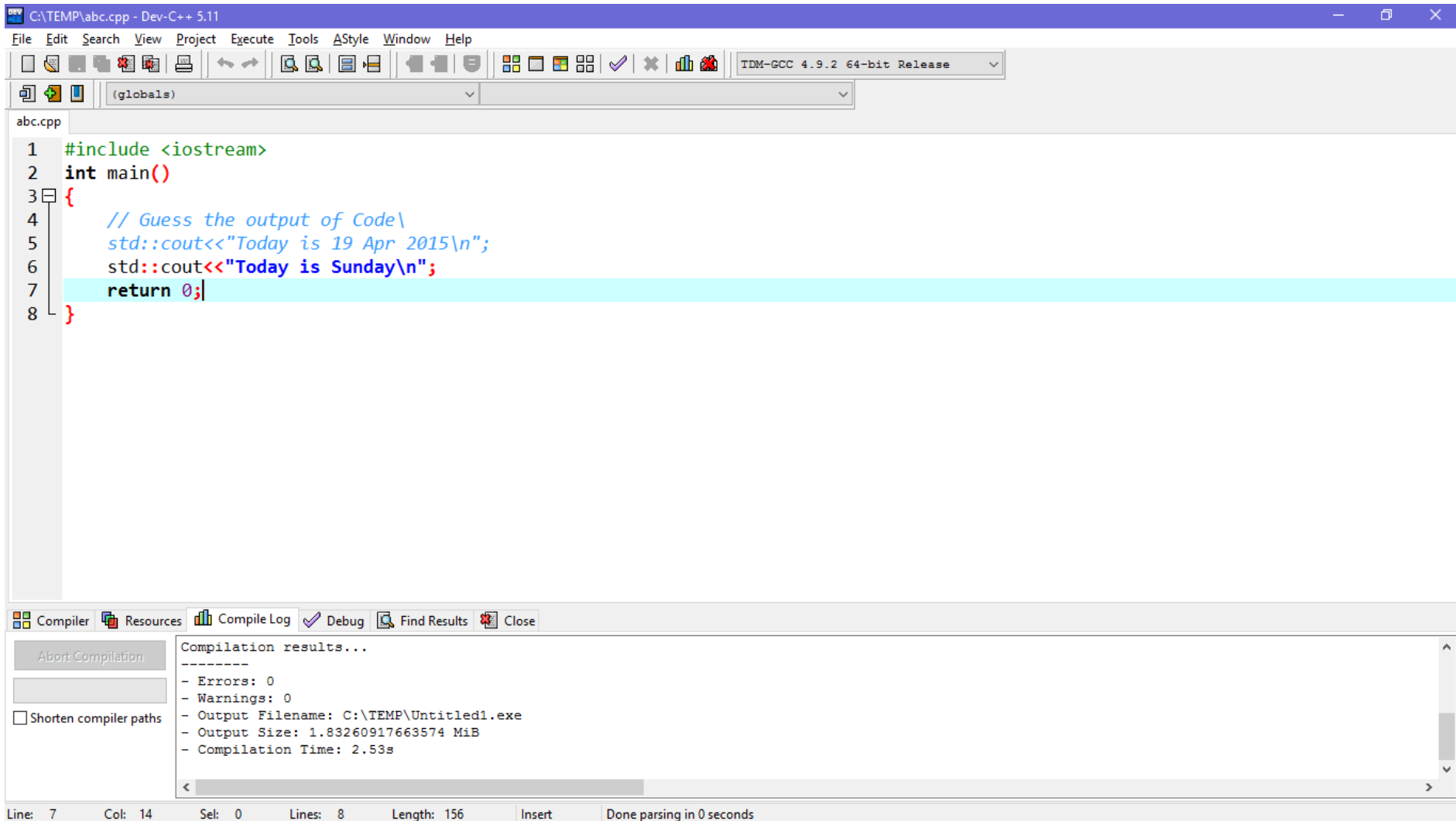
## 3.3 Công cụ Dev-C++

---

- Công cụ Dev-C++
- Hướng dẫn cơ bản
  - Bắt đầu vào chương trình
  - Viết mã
  - Dịch
  - Chạy
  - Sửa lỗi
- Một vài ví dụ đơn giản



# 3.3 Công cụ Dev-C++



The screenshot displays the Dev-C++ IDE interface. The main window shows a C++ source file named `abc.cpp` with the following code:

```
1 #include <iostream>
2 int main()
3 {
4     // Guess the output of Code\
5     std::cout<<"Today is 19 Apr 2015\n";
6     std::cout<<"Today is Sunday\n";
7     return 0;
8 }
```

The code is compiled successfully. The bottom status bar shows the compilation results:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\TEMP\Untitled1.exe
- Output Size: 1.83260917663574 MiB
- Compilation Time: 2.53s
```

The status bar at the bottom indicates the current cursor position: Line: 7, Col: 14, Sel: 0, Lines: 8, Length: 156, Insert, Done parsing in 0 seconds.

Phần 4

# Bài tập

# Bài tập

1. (về nhà) Cài đặt bộ công cụ Dev-C++ lên máy tính của bạn
  - Tải file cài đặt theo liên kết trong website bài giảng

2. Gõ thử chương trình sau:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Xin chao cac ban!" << endl;
}
```

3. Lưu chương trình thành file “**xinchao.cpp**”

4. Bấm F11 để dịch và chạy thử, sửa lỗi nếu có

# Thử nhập và chạy một số đoạn chương trình

---

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

int main() {
    cout << "Dong thu nhat" << endl;
    cout << "Dong thu hai" << endl;
}
```

# Thử nhập và chạy một số đoạn chương trình

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

int main() {
    int x;
    cout << "Nhap so can tinh: ";
    cin >> x;
    int m = x * x;
    cout << "Binh phuong la " << m << endl;
}
```

# Thử nhập và chạy một số đoạn chương trình

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

int main() {
    int x;
    cout << "Nhap so thu nhat:";
    cin >> x;
    int y;
    cout << "Nhap so thu hai:";
    cin >> y;
    int t = x + y;
    cout << "Tong hai so la " << t << endl;
}
```

# Thử nhập và chạy một số đoạn chương trình

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

int main() {
    double a, b, c;
    cout << "nhap a:"; cin >> a;
    cout << "nhap b:"; cin >> b;
    cout << "nhap c:"; cin >> c;
    double d = b*b - 4*a*c;
    double x1 = ( -b - sqrt(d) ) / (2 * a);
    double x2 = ( -b + sqrt(d) ) / (2 * a);
    cout << "x1 la " << x1 << endl;
    cout << "x2 la " << x2 << endl;
}
```