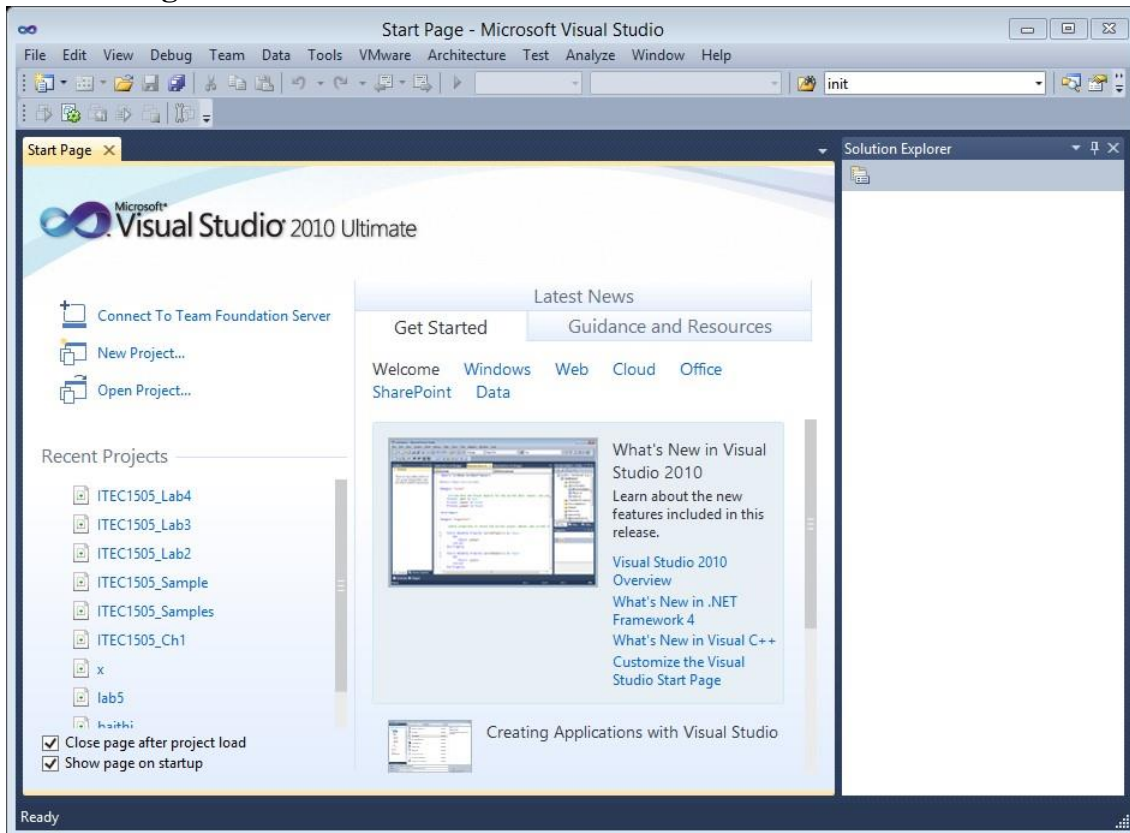


Phụ lục A

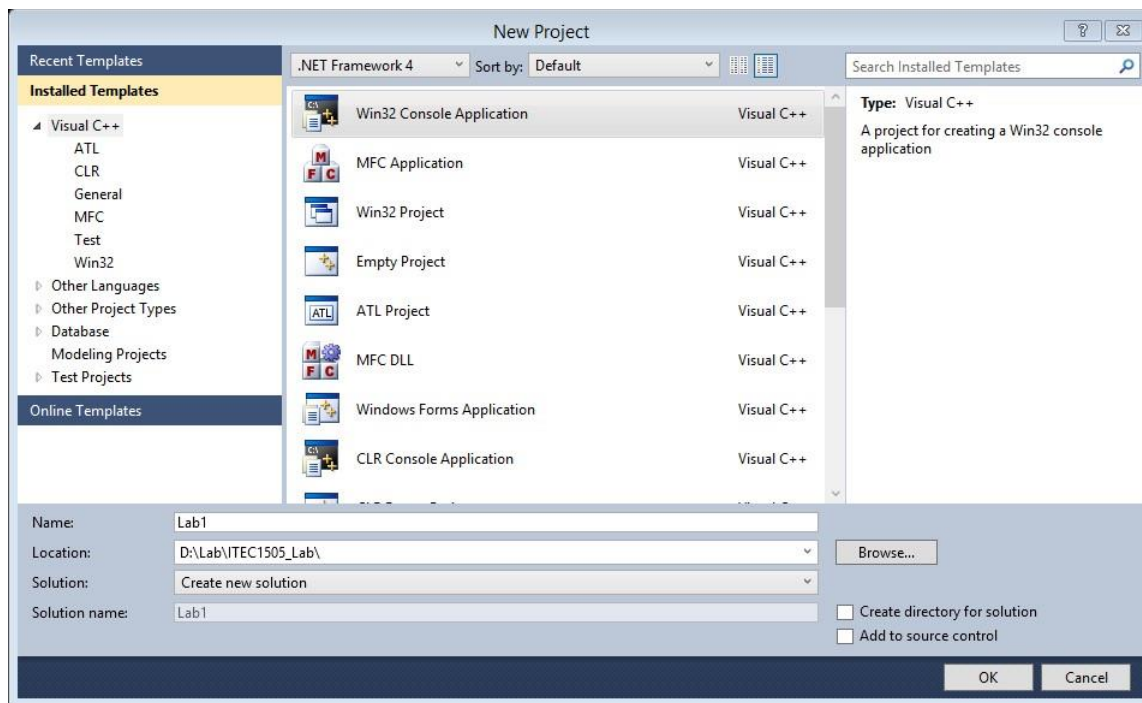
Tạo chương trình C++ trong Visual Studio 2010

1. Mở chương trình Visual Studio 2010.

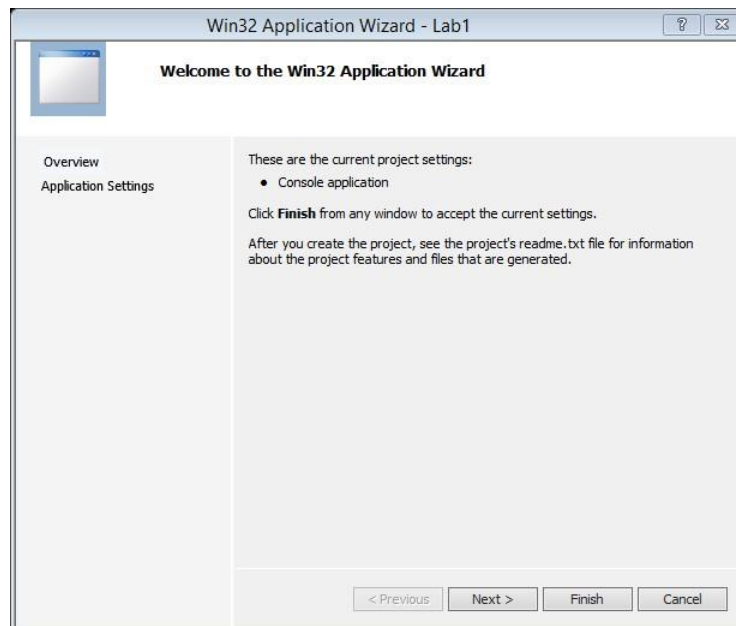


2. Mở menu File → New → Project...

- Chọn **Win32 Console Application**.
- Chọn thư mục lưu project trong **Locations**.
- Gõ vào tên của project trong ô **Name**.
- Click **OK**.

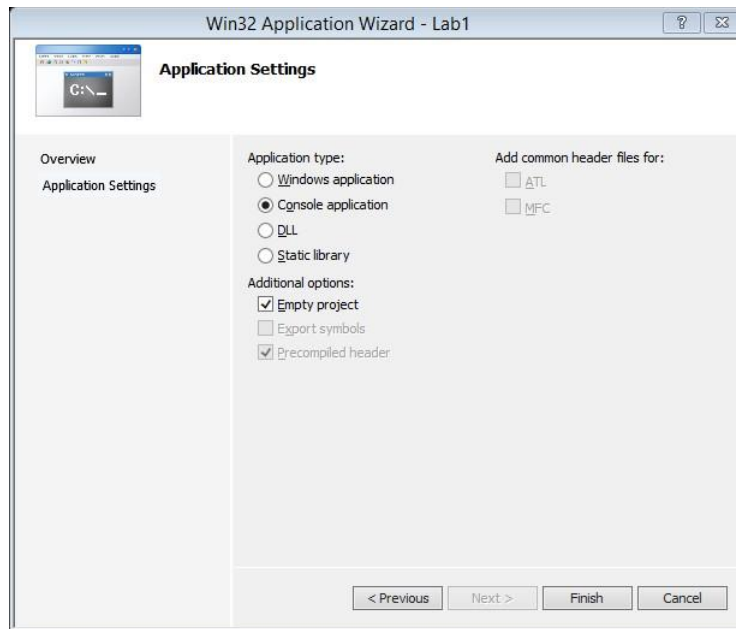


3. Một cửa sổ xuất hiện



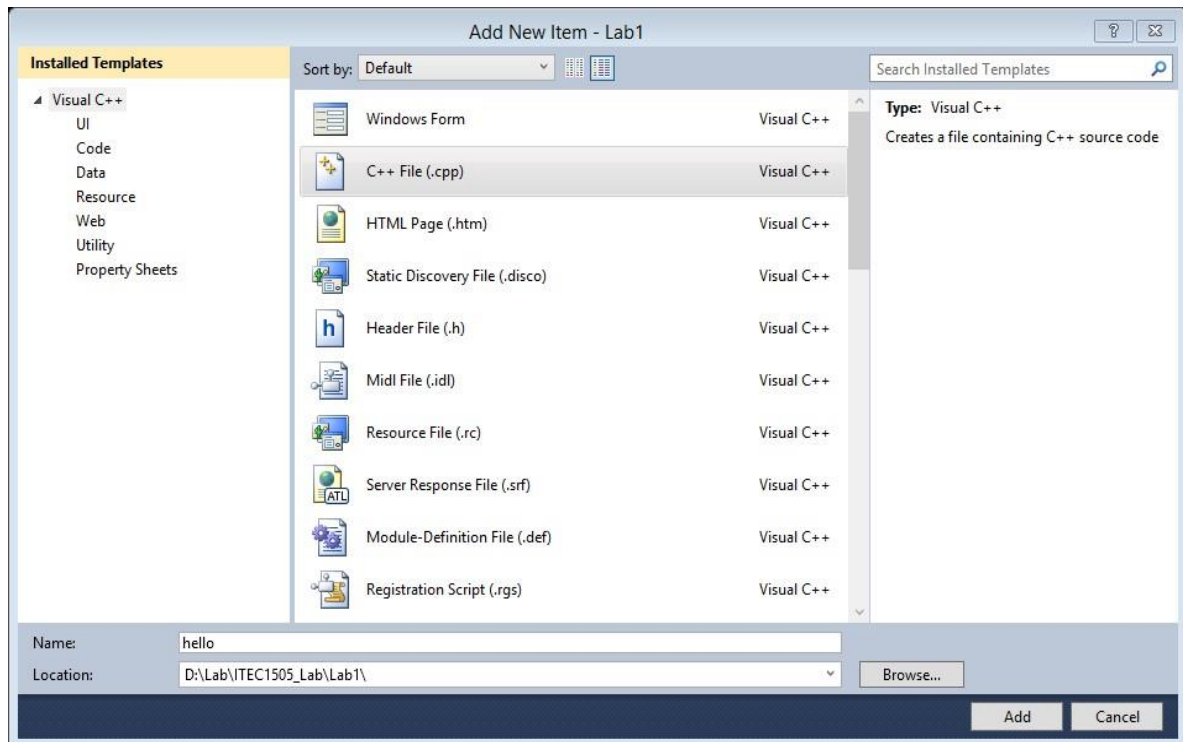
- Tên cửa sổ là tên của project.
- Click **Next**. Một cửa sổ khác xuất hiện.

4. Đánh dấu chọn **Empty Project** và click **Finish**.

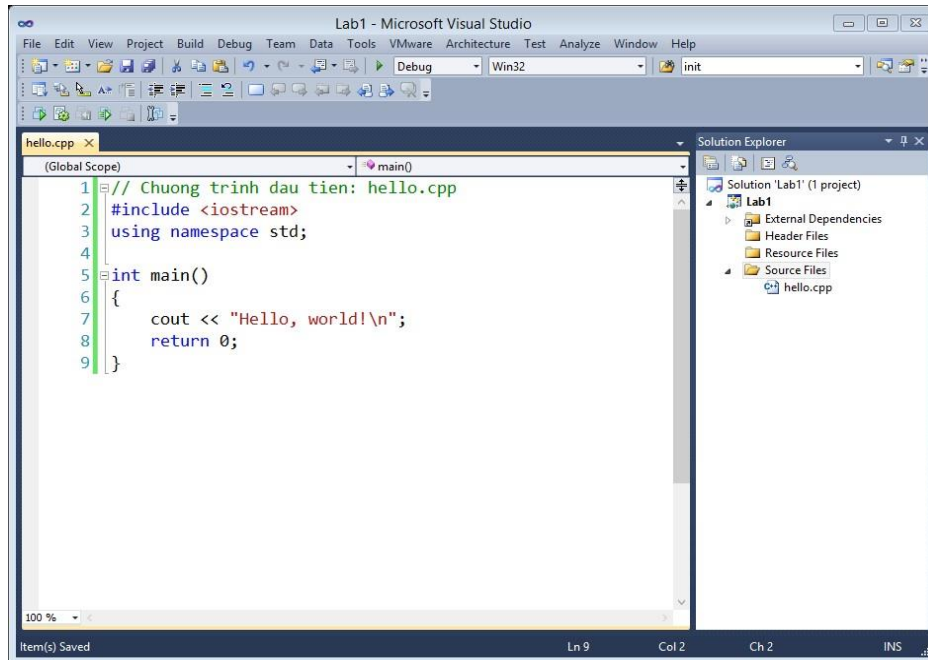


5. Tạo và thêm file nguồn vào project để gõ chương trình.

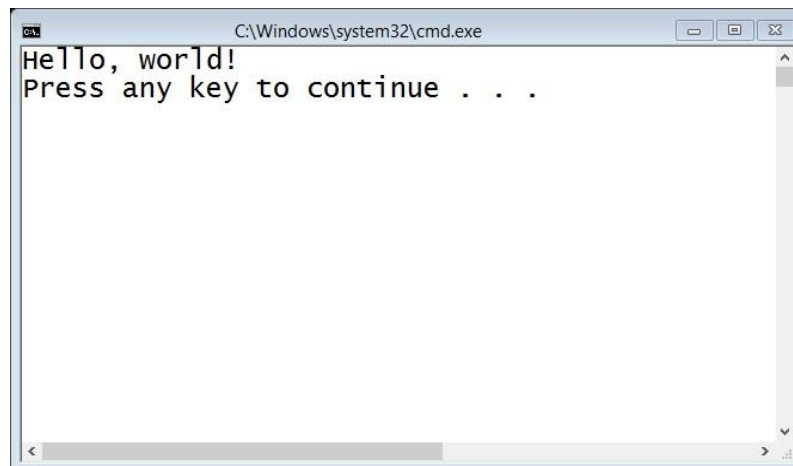
- Click **Project** → **Add New Item...**
- Chọn **C++ File (.cpp)**.
- Gõ vào tên file trong ô **Name**.



6. Gõ vào chương trình.



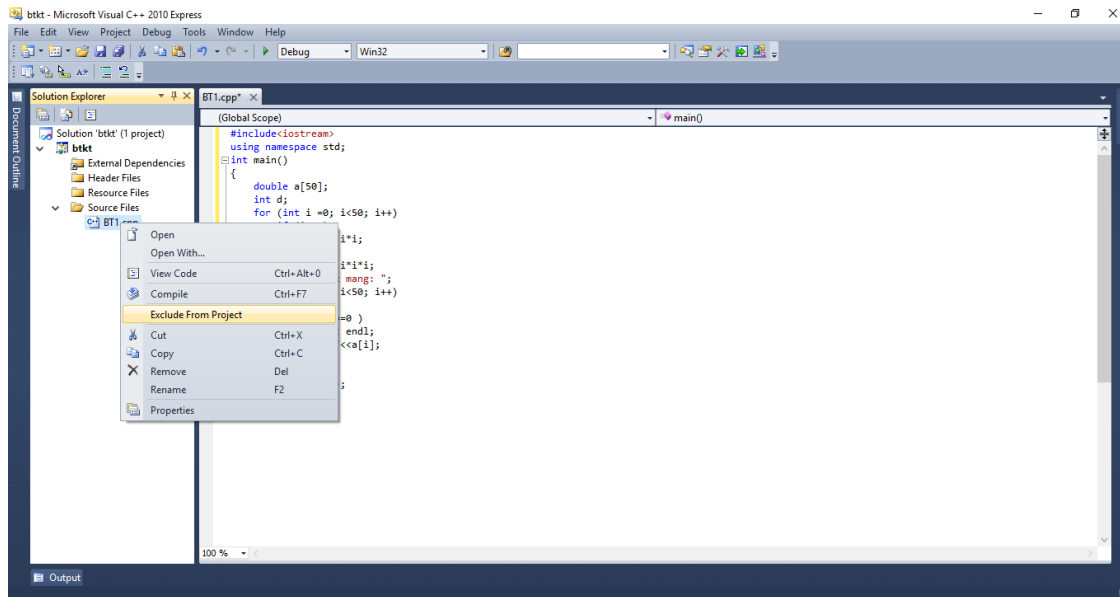
7. Biên dịch và chạy chương trình bằng cách chọn Debug → Start Without Debugging, hoặc nhấn CTRL+F5.



8. Di chuyển file source code đã làm việc xong ra khỏi Project:

Một project trong Visual Studio sẽ chỉ chấp nhận một hàm main duy nhất trong quá trình làm việc. Đồng nghĩa với việc nếu bạn viết nhiều chương trình (mỗi chương trình bao gồm một hàm main) thì Project sẽ gây lỗi biên dịch.

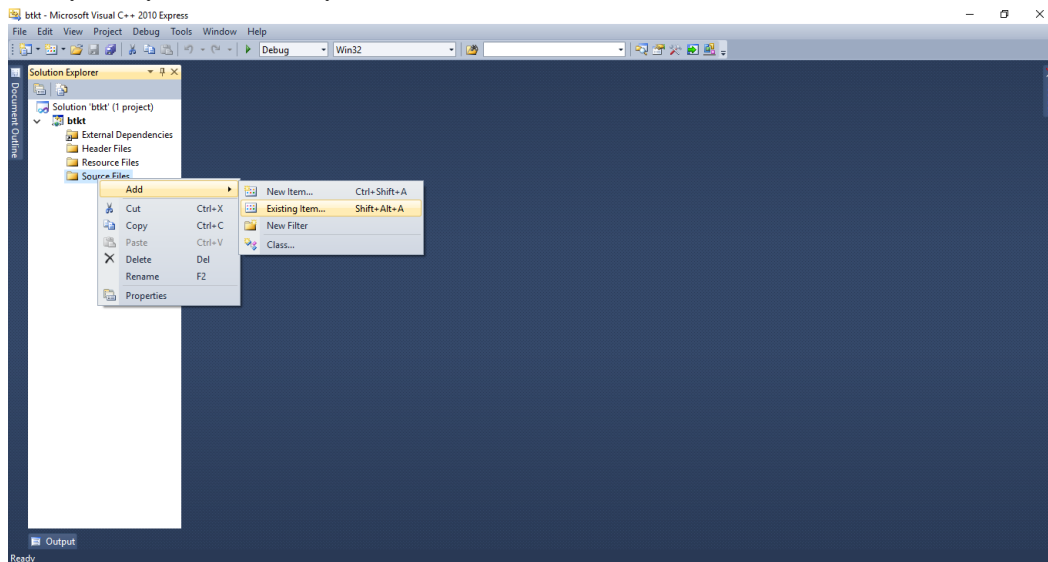
Vì vậy, khi đã kết thúc một chương trình (file source code .cpp) nào đó xong thì bạn phải di chuyển file này ra khỏi Project để tránh gây lỗi biên dịch. Thực hiện việc này bằng cách **Relick vào file** (Trên thanh Solution Explorer) và chọn **Exclude From Project**. File này thực tế không di chuyển khỏi Project, mà chỉ không tham gia vào quá trình biên dịch. Khi bạn mở Project trên Window Explorer thì File này vẫn tồn tại.



9. Mở 1 file source code (.cpp) đã có vào làm việc:

Khi project cần mở lại một file source code (.cpp) để làm việc thì Project đó phải không có hàm main nào đang hiện hành. Sau đó thực hiện:

- **Relick vào Folder Source Files** trên Solution Explorer,
- Chọn **Add -> Existing Item...**
- Lựa chọn file rồi chọn **Add**.



Phụ lục B

Một số hàm thư viện

<cmath>

<code>double sqrt(double x)</code>	Căn bậc hai của x
<code>double pow(double x, double y)</code>	Lũy thừa, x^y .
<code>double sin(double x)</code>	$\sin(x)$ (đơn vị x là radians).
<code>double cos(double x)</code>	$\cos(x)$ (đơn vị x là radians).
<code>double tan(double x)</code>	$\tan(x)$ (đơn vị x là radians).
<code>double log10(double x)</code>	$\log_{10} x, x > 0$.
<code>double fabs(double x)</code>	Trị tuyệt đối, $ x $.

<cctype>

<code>bool isalpha(char c)</code>	Kiểm tra là ký tự chữ.
<code>char isalnum(char c)</code>	Kiểm tra là ký tự chữ hoặc chữ số.
<code>bool isdigit(char c)</code>	Kiểm tra là chữ số.
<code>bool isspace(char c)</code>	Kiểm tra là khoảng trắng.
<code>bool islower(char c)</code>	Kiểm tra là chữ thường.
<code>bool isupper(char c)</code>	Kiểm tra là chữ in.
<code>char tolower(char c)</code>	Trả về ký tự chữ thường của c .
<code>char toupper(char c)</code>	Trả về ký tự chữ in của c .

<cstdlib>

<code>int abs(int x)</code>	Trị tuyệt đối, $ x $.
<code>void exit(int n)</code>	Thoát khỏi chương trình, n là mã trạng thái.
<code>int rand()</code>	Tạo số nguyên ngẫu nhiên.
<code>int srand(int n)</code>	Dùng với <code>rand()</code> .

<string>

<code>getline(cin, string s)</code>	Lưu trữ một luồng nhiều ký tự vào cho bộ nhớ (kể cả khoảng trắng) với tên gọi s .
<code>int s.length()</code>	Thuộc tính trả về chiều dài của chuỗi s

Phụ lục C

Cách debug chương trình trong C++ trong Visual Studio 2010

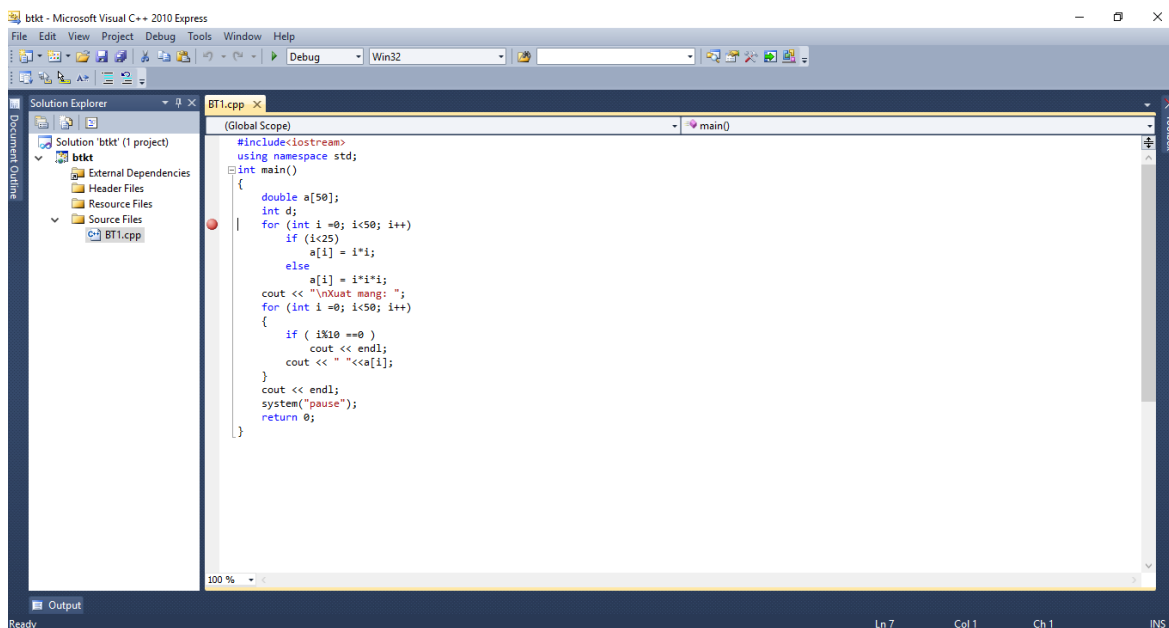
Đây là tính năng gỡ rối (debug) được hầu hết các IDE hỗ trợ. Khả năng này giúp chạy từng dòng lệnh (step by step) của chương trình. Dựa vào đó người lập trình có thể biết được sự thay đổi giá trị của các biến, kiểm tra kết quả từng giai đoạn trong chương trình.

Trình tự thực hiện debug:

- Đặt Breakpoints.
- Vào menu Debug -> Start Debugging hoặc nhấn phím F5. Thực hiện các bước Debugging.
- Quan sát các kết quả ở màn hình.
- Thoát khỏi quá trình debugging.

1. **Breakpoints:** là vị trí mà bạn muốn chương trình sẽ dừng lại để xem xét sự thay đổi của các biến qua từng dòng lệnh. Dựa vào đó có thể phát hiện ra vị trí dòng code bị lỗi.

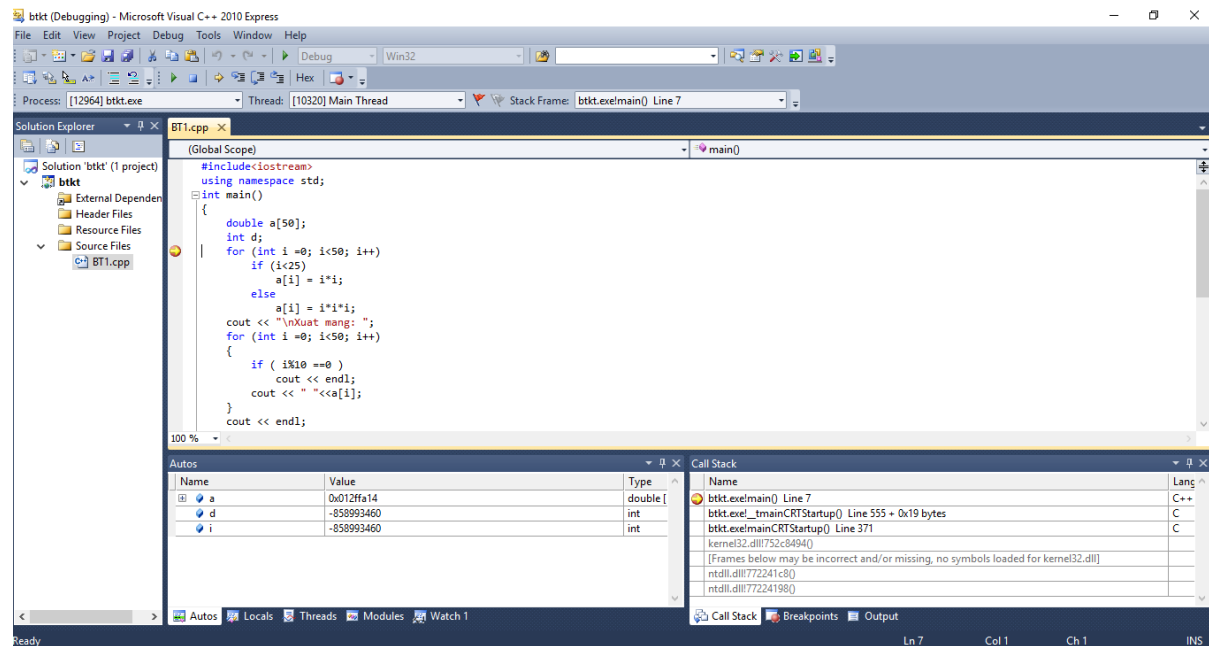
Breakpoint được kí hiệu bằng chấm tròn màu đỏ ở đầu dòng code. Để tạo ra một breakpoint, ta click chuột vào đầu dòng code:



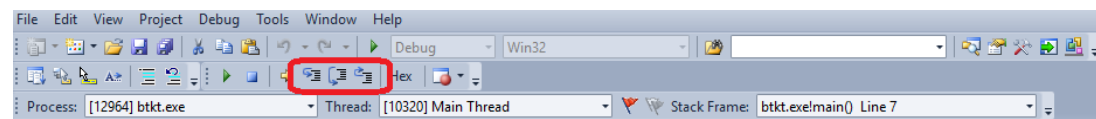
Để huỷ breakpoint, chỉ cần click chuột vào breakpoint đó một lần nữa. Ngoài ra các bạn cũng có thể tạo/huỷ breakpoint bằng phím F9.

2. Debugging:

Chương trình sẽ dừng hiển thị trên Console ở vị trí dòng code mà bạn đã đặt Breakpoints.



Bạn cần quan tâm đến thanh công cụ Debug bên trên:



Trong đó:

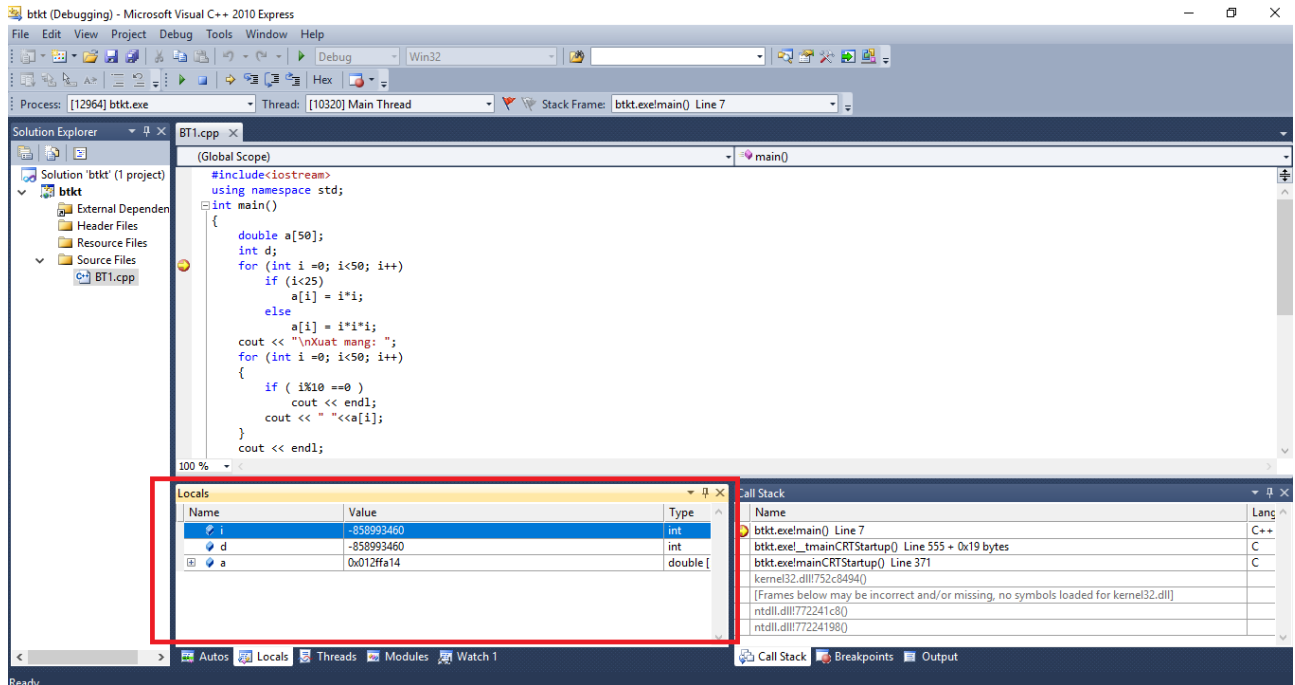
- **Step Over:** Chạy step by step, lướt qua hàm (chỉ nhận giá trị return của hàm).
- **Step Into:** Chạy step by step, đi vào nội dung của các hàm con.
- **Step Out:** “Nhảy” đến breakpoint kế tiếp. Nếu không còn breakpoint nào thì sẽ kết thúc debug. Ngoài ra nó còn có chức năng chạy lướt qua hàm con hiện tại.

Các phím tắt cần thiết:

- **F5:** Bắt đầu quá trình Debug.
- **Shift + F5:** Thoát Debug.
- **Ctrl + F5:** Chạy chương trình không dùng công cụ Debug
- **F9:** Tạo/huỷ một breakpoint.
- **F10:** Step Over.
- **F11:** Step Into.

- **Shift + F11: Step Out.**

3. Quan sát kết quả: Sự thay đổi giá trị các biến sau từng đoạn code sẽ được hiển thị ở màn hình Locals:



4. Dừng debugging: Menu Debug -> Stop Debugging.

Tài liệu tham khảo

- [1] Diane Zak. An Introduction to Programming with C++, Eighth edition, Cengage Learning, 2016.
- [2] Paul Deitel & Harvey Deitel. C++ How to Program, Eighth edition, Pearson, 2012.
- [3] Bjarne Stroustrup. The C++ Programming Language, Third Edition, Addison-Wesley, 1997.
- [4] Bjarne Stroustrup. Programming - Principles and Practice Using C++, Addison-Wesley, 2008.
- [5] Herbert Schildt. The Complete Reference C++, Fourth edition, McGraw Hill, 2003.