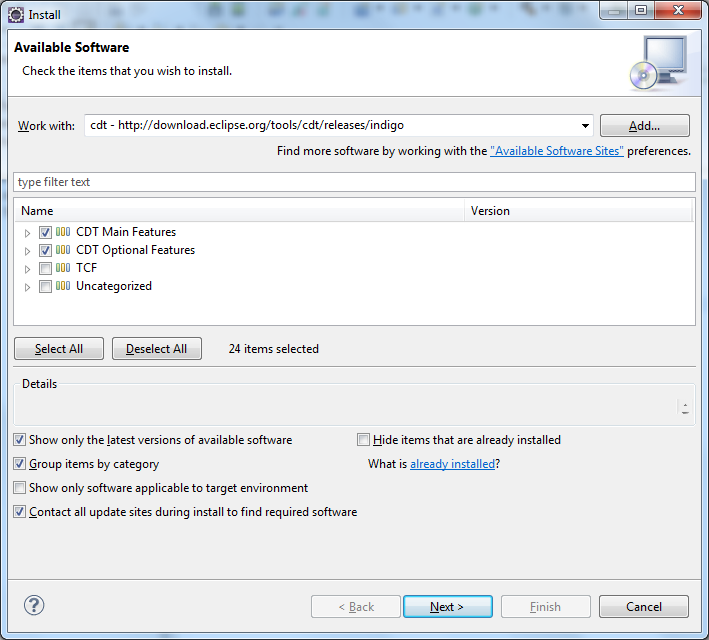
Setup môi trường cho eclipse

# Cài đặt CDT Plugin cho eclipse

Đây là bộ công cụ chứa các công cụ IDE cho C/C++



Link đến gói cài đặt trên là: <http://download.eclipse.org/tools/cdt/releases/indigo>

Khi đã cài đặt xong thì máy đã có IDE lập trình C/C++. Nhưng để Build và Link được ra file thực thi thì cần cài đặt **compiler** vì bản thân CDT không có sẵn compiler.

# Cài đặt compiler

## Host và Targer là môi trường Windows

Khi lập trình thì cần có hai khái niệm quan trong trong phát triển và biên dịch chương trình.

Khái niệm host là chỉ máy dùng để lập trình.

Khái niệm target chỉ máy sẽ chạy chương trình mà ta viết.

Để viết một ứng dụng C/C++ chạy trên windows ta có thể dùng compiler là Microsoft Visual C++ và Microsoft Windows SDK. Đã được cài đặt sẵn nếu máy có cài Visul Studio. Hoặc cũng có thể dùng **MinGW** là "Minimalist GNU for Windows", là một bản port của GNU trên môi trường Windows, dùng để biên dịch ứng dụng chạy trên Windows, nó sử dụng các Windows API được cung cấp từ HDH và không sử dụng bất kỳ DLL của bên thứ 3 nào.

## Host và Targer là môi trường Linux

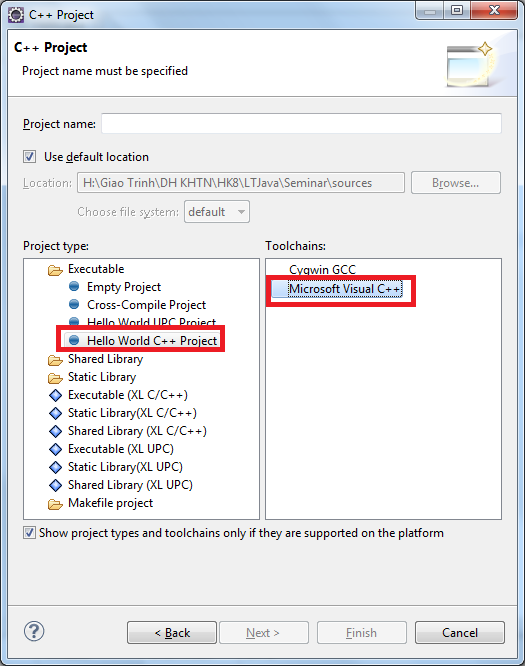
Trên Linux thì dùng gcc làm compiler. Nên cài đặt và cấu hình không phức tạp như trên windows.

# Kiểm tra môi trường C/C++ bằng ứng dụng Helloworld

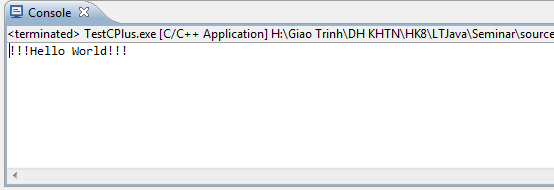
Vào menu File🡪 New 🡪 Other… chọn C++ Project

Nhập vào tên project và chọn ví dụ hello world mẫu.

Toolchains chọn compiler dùng biên dịch.



Sau đó chọn Run As 🡪 Local C++ Application nếu chạy thành công thì output sẽ xuất ra helloworld:

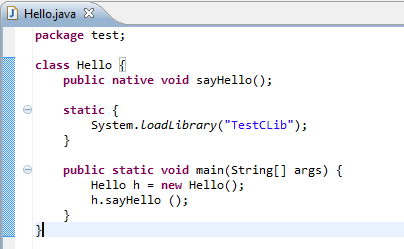


Thực tế có thể gặp vài lỗi nhỏ. Chi tiết mời xem file video demo đính kèm.

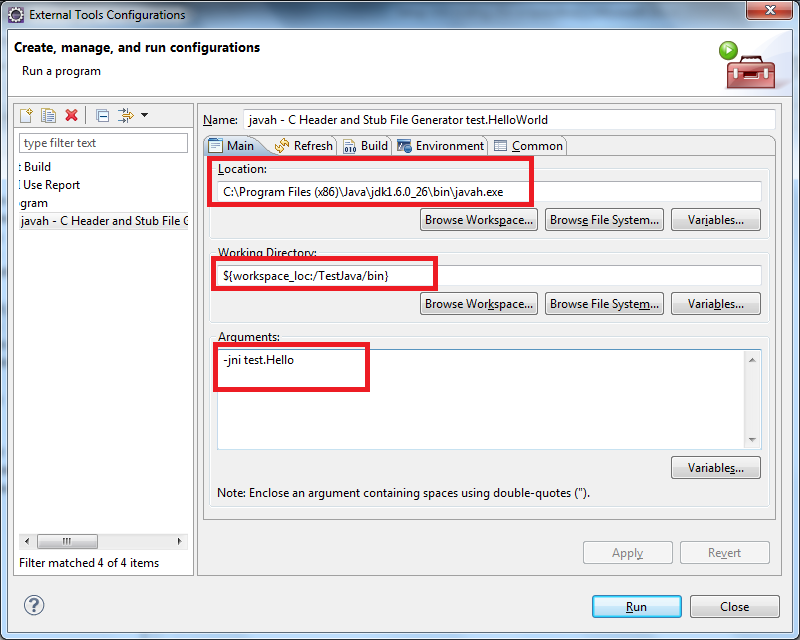
# Ứng dụng Java kết hợp native code đầu tiên

Để minh họa cũng như nắm bắt được cách để kết hợp native code trong chương trình java chúng ta hãy cùng nhau làm một ví dụ helloJNA cơ bản sau.

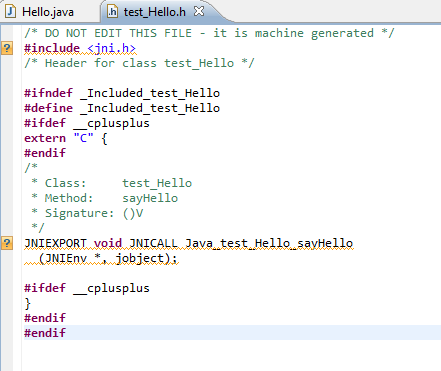
Về cơ bản chúng ta sẽ tạo 2 project trong eclipse. Một Java Project chứa hàm main sẽ gọi một hàm native xuất ra chữ HelloJNA.



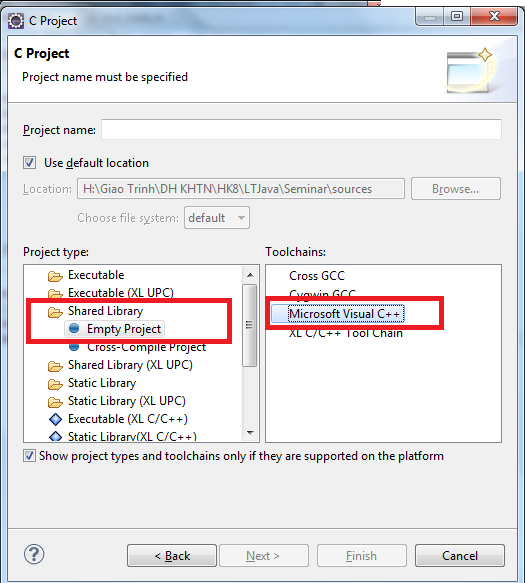
Tạo file header bằng javah (kèm theo jdk), là file chứa prototype các hàm cần cài đặt bằng c++. Vào menu Run 🡪 External tool 🡪 External tool configuration… tạo một config mới tham số chủ yếu là đường dẫn đến file javah.exe, nơi lưu file .h, lớp java cần tạo header.



Sao khi chạy config ta vừa add vào thì trong thư mục bin của project sẽ có file test\_Hello.h với nội dung như sau:



Tiếp theo là tạo project thứ 2 là một C Project (dạng library) cài đặt hàm sayHello. Chọn loại project là Shared Library (khi biên dịch sẽ ra file dll).



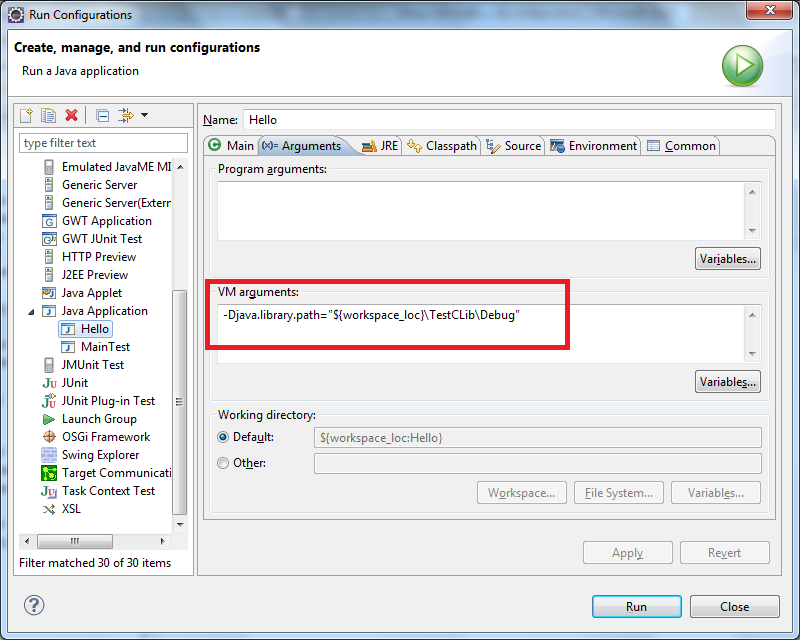
Trong thư mục src của project C thêm file test\_Hello.c để cài đặt hàm sayHello trong file .h trên.



Chúng ta có thể chép file .h từ project java sang project c này hoặc cũng có thể chỉ đường dẫn tương đối đến nó.

Để biên dịch được Project C này cần cấu hình include của C/C++ compiler đến thư mục include và include\win32 của JDK và library thêm đường dẫn đến file kernel.obj của Microsoft Windows SDK. Chi tiết xem file video demo.

Khi đã biên dịch thành công project C ta sẽ tạo cấu hình để chạy project java. Điểm mấu chốt là để JVM load được file dll của project C. Có thể làm việc này bằng 2 cách. Cách 1 là thêm vào biến môi trường Path đường dẫn đến thư mục output của project C. Cách 2 là truyền tham số cho JVM.



Cuối cùng chỉ là run Java Application theo cấu hình vừa mới tạo.