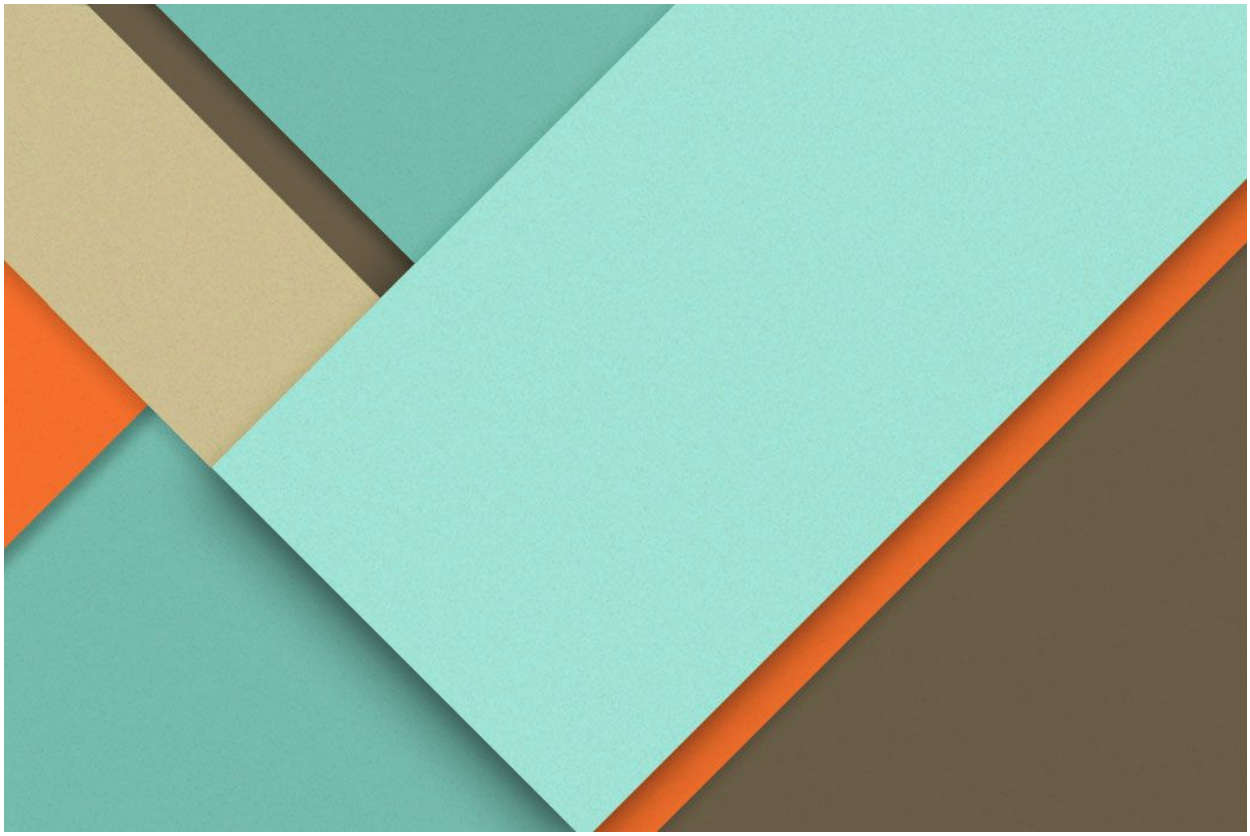

Warn if virtual calls are made from constructors or destructors

04.05.2017

Tushar Khurana
New Delhi, India

Overview

Implement a path-sensitive checker that warns if virtual calls are made from constructors and destructors, which is not valid in case of pure virtual calls and could be a sign of user error in non-pure calls.

The current virtual calls checker, implemented in VirtualCallChecker.cpp, needs to be re-implemented in a path-sensitive way. The lack of path-sensitive reasoning may result in false positives in the inter-procedural mode, which is disabled now for that reason. The false positives could happen when a called function uses a member variable flag to track whether initialization is complete and relies on the flag to ensure that the virtual member function is not called during initialization. Further, the path diagnostic should be used to highlight both the virtual call and the path from the constructor. Last, we will need to evaluate if the warning should be issued for both calls to pure virtual functions (which is always an error) and non-pure virtual functions (which is more of a code smell and may be a false positive).

In this project the solution will be to use clang's LibTooling library.

Goals

Issue a warning for calls to virtual functions from inside of constructor or destructor.

Specifications

Here, for the static analyzer to detect whether there is a virtual call to a function being made, we will use the Clang LibTooling and LibClang library to get the ast-dump of the source code, then using that ast-dump and ASTcontext we will find out where are the bodies of constructors and destructors in a translation unit, and then for each line in their bodies check whether that is a function call to a virtual function, if yes, we will get the appropriate line number in the source code corresponding to that virtual function call in the constructor and destructor, and then when the bodies of all constructors and destructors have been checked, we will print out the warnings for those line numbers along with the virtual function name called on those line numbers.

Milestones

I. 10 July 2017

Prototype application ready for detecting function calls in any source code.

II. 15 July 2017

Application for detecting constructor and destructor calls ready in any source code.

III. 25 July 2017

Complete application ready for detecting virtual function calls from within constructors and destructors using ASTContext and ast dump of a source code which will issue a warning where virtual function call is found inside constructor or destructor.

IV. 27 July 2017

Testing completed and final application ready.

Note:

This proposal is made at 4 May 2017 at 5:30AM IST, just hours before the GSoC result, because at first(around 3 April 2017) I overlooked this task by LLVM Clang, hence this proposal might look too small, but feel free to contact me for more details regarding this project and the other project for Clang Bash completion that I submitted :-).