1 Missing semantic checks

If a compiler neglects to make semantic checks, problems can occur later. The Cool compiler has an undocumented option `-f` that tells the compiler to muffle all semantic errors. It is used as in

```
coolc -f bad.cool
```

For each of the following errors, try out the `-f` option on programs that commit the error. Make sure you compile without `-f` as well to make sure you don’t have multiple errors in the file! Explain what results you get. For each check, classify the check as one of the following: (1) the check is actually unnecessary to ensure safety, or (2) removing the check may cause the code generator to crash, or (3) removing the check may cause the generated code to be malformed for SPIM so that execution cannot start, or (4) removing the check permits the generated code to do something illegal. For (1), you should try to convince me that the check is unnecessary. For (2), (3) or (4) a simple example demonstrating the problem suffices.

1. The `Main` class must have no parameters.

2. You are not allowed to have two cases for the same type.

3. You are not allowed to have two formals of the same name.

4. A derived class may not “shadow” attributes of the superclass.

5. You may not inherit from class `String`.

6. The return type of a method must conform to the return type of a method that it overrides.

7. You may not overload methods.

8. The `Null` type is not a subtype of `Boolean`.

9. You are not allowed to add strings: "hello" + "world"

10. A method marked `override` must override a method.

Put all Cool files and an overall explanation (homework4.txt) in your homework4 directory of your AFS volume.

2 Testing

Inheritance and overriding can expose compiler bugs. Write tests in `good.cool` and `bad.cool` (put in homework4/) and use the test5 script as with previous assignments.
3 Multiple Inheritance

C++ provides virtual and non-virtual multiple inheritance. Java only permits multiple inheritance of “interfaces.” Is Java’s multiple inheritance like C++ virtual multiple inheritance or the nonvirtual kind? Explain! What are the pitfalls of full multiple inheritance in C++? How does Java avoid them, even with Java 8’s “default” methods?

4 Alternatives to Multiple Inheritance

Researchers have proposed “traits” and “mixins” as way to give expressive power while avoiding the problems of multiple inheritance.

Read and cite at least three research papers on such topics. Explain the main ideas and how they permit useful expressiveness while avoiding pitfalls of multiple inheritance.