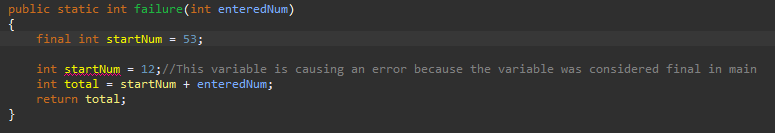
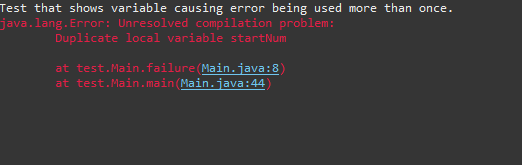
The scope rules of modern programming languages are rooted in rules developed by the earliest block structured languages like Algol. Adding object orientation to languages adds one more layer of complexity to this issue.

Let's consider Java, a language with which everyone should be familiar, and explore its scope rules. One aspect of the scope rules of any language is when data or methods can have the same name. Give examples in Java to illustrate a case where declaring two local variables with the same name in same method is permitted and one where it is prohibited. Are the rules governing redeclaration of local names the same as those governing redeclaring local names that rename class-level names?

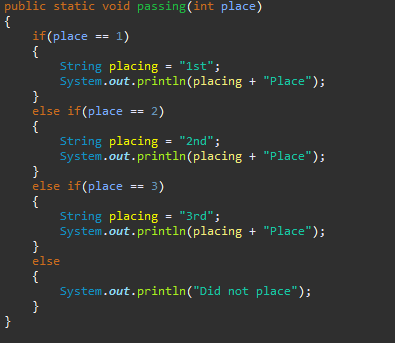
One consequence of scope rules in most languages is that forward references (referring to names before they are declared) are prohibited. Are such forward references always prohibited within a single Java class? Are forward references of local names always prohibited within a single method? If not, provide an example that demonstrates your claim.

In the method *failure(),* the variable “startNum” is set to final in the beginning of the method. When the program tries to call the variable again, the compiler catches an error.





In the method *passing(),* the variable “place” is used multiple times in the *if* statement, since the variable will only be called upon if the argument is met, the compiler will compile.





public class Main

{

public static int failure(int enteredNum)

{

//final int startNum = 53;

int startNum = 12;//This variable is causing an error because the variable was considered final in main

int total = startNum + enteredNum;

return total;

}

public static void passing(int place)

{

if(place == 1)

{

String placing = "1st";

System.***out***.println(placing + "Place");

}

else if(place == 2)

{

String placing = "2nd";

System.***out***.println(placing + "Place");

}

else if(place == 3)

{

String placing = "3rd";

System.***out***.println(placing + "Place");

}

else

{

System.***out***.println("Did not place");

}

}

public static void main(String[] args)

{

//test that show String variable "place" being used in more than once

System.***out***.println("Test that shows variable being used more than once.");

*passing*(2);

//test that show variable "startNum" being used in more than more place

System.***out***.println("Test that shows variable causing error being used more than once.");

System.***out***.println(*failure*(89));

}

}//end public class Main