Andreas Landgrebe
September 10, 2015
Lab 2

3.Considering the following JavaScript program - what value is printed by the final line - is it "hello" or is it "10"?

The values that are printed by this JavaScript program is "10" and then the final line is "hello".

a. Explain how JavaScript's "function scope" rule is interpreted.

JavaScript has two scopes: global and local. If a variable is declared outside the functions, then it is global scope and one is able to access this variable from anywhere in the source code. Each functions has its own scope, and any variables declared within that functions can only be accessed from that functions and any nested functions. Because local scope in JavaScript is created by functions, its also called function scope.

b. State whether or not JavaScript requires "declare before use" for variables

JavaScript does not require declare before use for variables as long as it is assigned somewhere in the source code.

5. Look at the stack structure in Java
In your document, "Draw" the portion of the frame containing the parameters and local variables of function f

1. load the int value i in frame location 1
2. load the int value j in frame location 2
3. add i and j together
4. store int value into variable 9
5. load the double a from local variable 3
6. load the double b from local variable 5
7. multiple the two doubles
8. store the double value prod into local variable 10
9. load the int (or char) p from local variable 7

10. store the int (or char) max in local variable 12
11. load the int (or char) p from local variable 7
12. load the int (or char) q from local variable 8
13. if the int (or char) value p of local variable 7 is less than the int (or char) q value of local variable 8, branch to instruction at branchoffset.
14. load the int (or char) q value from local variable 8
15. store the int (or char) max value from local variable 12
16. return

6. A stack machine computation
"Draw" the frame for f, then "draw" the contents of the stack after each line of bytecode in function f.

1. load the int value x 10 into local variable 1
2. load the int value y 20 into local variable 2
3. add x and y together to get 30
4. load the int value x 10 into local variable 1
5. load the int value y 20 into local variable 2
6. add x and y together to get 30
7. multiply x and y together to get 900
8. load the int value x into local variable 1
9. load the int value y into local variable 2
10. add x and y together to get 30
11. load the int value x into local variable 1
12. load the int value y into local variable 2
13. add x and y together to get 30
14. multiply x and y together to get 900
15. add x and y together to get 1800
16. store the final value of 1800 into local variable 3

One more look at optimization.
How could this bytecode be optimized, i.e., shortened to fewer instructions? Answer this question, explaining and showing your optimized bytecode, in your document.

This bytecode is as short as it is due to the fact that you would need to load the int values x and y each time in order to do any adding and multiplying. However, you can optimize this code. The following will optimize the code

1. iconst_2
2. iload_1
3. iconst_2
4. ixor
5. iconst_2
6. iload_1
7. imul
8. iload_2
9. imul
10. iadd
11. iload_2
12. iconst_2
13. ixor
14. iadd
15. imul
16. istore_3
17. return

Instead of loading the variables of x and y constantly, it would be more effective to use the iconst instruction to load the int value of 2 to be used to calculate. Stack3.java is shown in the repository to show the modified and improved code that will give you the same result by a different way.