CSE 6341, Written Assignment 4  
Due Thursday, April 14, 9:30 am (8 points)

This assignment contains 3 questions, for a total of 8 points. Your submissions should be uploaded via Carmen. Create your answers using latex, Word, or plain text. If the answers are too difficult to write in an editor, feel free to write it by hand on a piece of paper and then upload a legible scan or photo of it.

You can submit up to 24 hours after the deadline; if you do so, your score will be reduced by 10%. If you submit more than 24 hours after the deadline, the submission will not be accepted.

Q1 (2 points): In class we briefly mentioned the use of operational semantics to prove the correctness of compiler transformations. As an example, consider the following code transformation:

```
if (be) then { x = e1; } else { y = e2;);
x = e1;
```

is transformed into

```
if (be) then { x = e1; } else { y = e2; x = e1;};
```

Here be is some Boolean expression and e1 and e2 are some arithmetic expressions without side effects. Show an example of concrete code for which this transformation is not semantics-preserving.

Q2 (3 points): Consider the control-flow graph (CFG) shown below. For each node, describe the immediate dominator of that node. Based on this information, describe all back edges in the graph. For each back edge, describe its natural loop – specifically, (1) the header of that loop, and (2) the set of nodes in that loop.

Q3 (3 pts): Suppose node 2 from the above CFG contains instruction x = 3.14 * 3.14. Further, suppose that x is not assigned in any other instruction in the CFG. Can a compiler move this loop-invariant instruction from node 2 to node 1 in the CFG? Justify your answer, based on the correctness discussion in class.