

Creativity Challenge – Code Review Template

Coder Name: Nolan Downey

Reviewers Names: Kelly Buchanan, Connor Ruff

Creativity Challenge Number:

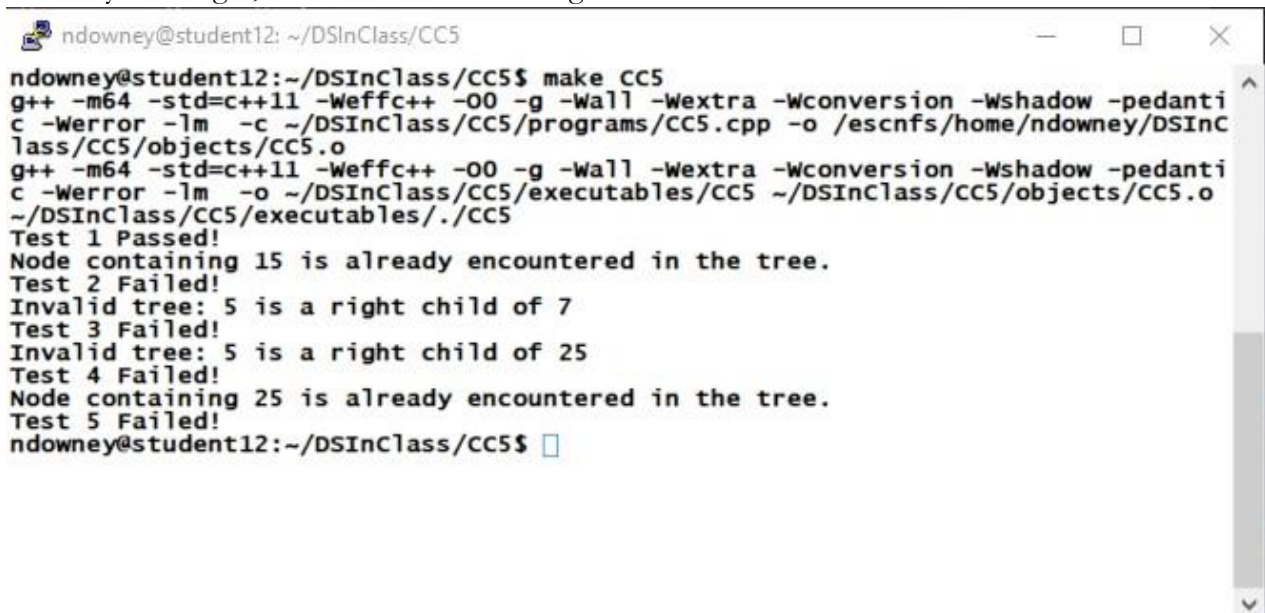
Part 1 - Correctness and Defects – 150 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

Does the program do what you expect?

- If the code compiles and produces the correct output, to earn the 150 points:
- Describe in detail why your reviewers believe it works
 - Describe their assessment of your proposed solution based on problem statement and the result generated just for correctness (efficiency and memory are in Parts 2 and 3)
 - Show the code run for several test cases
- If the code does not compile or produce the correct output, to earn the 150 points:
 - Describe the currently existing compiler/Makefile linker errors, if any exist.
 - Describe the code segments that are potentially contributing to these errors.
 - Describe the currently incorrect outputs and logical defects, if any exist
 - Describe the code segments that are potentially contributing to these errors.

Reviewer Comments:

- After investigation of my code by my reviewers, we determined that my code produces the correct output. For this specific Creativity Challenge, Dr. Morrison gave us a program that tested the method/function that we wrote with 5 different types of trees. We were expected to have the code that we wrote produce a pass/fail output based on these tests. My code produces the correct output based on the trees from the creativity challenges, as seen below in the image.



```
ndowney@student12: ~/DSInClass/CC5
ndowney@student12:~/DSInClass/CC5$ make CC5
g++ -m64 -std=c++11 -Weffc++ -O0 -g -Wall -Wextra -Wconversion -Wshadow -pedanti
c -Werror -lm -c ~/DSInClass/CC5/programs/CC5.cpp -o /escnfs/home/ndowney/DSInC
lass/CC5/objects/CC5.o
g++ -m64 -std=c++11 -Weffc++ -O0 -g -Wall -Wextra -Wconversion -Wshadow -pedanti
c -Werror -lm -o ~/DSInClass/CC5/executables/CC5 ~/DSInClass/CC5/objects/CC5.o
~/DSInClass/CC5/executables/./CC5
Test 1 Passed!
Node containing 15 is already encountered in the tree.
Test 2 Failed!
Invalid tree: 5 is a right child of 7
Test 3 Failed!
Invalid tree: 5 is a right child of 25
Test 4 Failed!
Node containing 25 is already encountered in the tree.
Test 5 Failed!
ndowney@student12:~/DSInClass/CC5$
```

Part 2 – Performance – 125 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

The team's assessment of the run-time efficiency of your proposed solution.

- Does this program use the best Data Structure for runtime given the problem constraints?
 - If your teams believe so, state why.
 - If your team believes not, state why not.
- Does this program run in the most efficient manner possible given the available resources? (i.e., not using unnecessary loops and complex approaches when a Data Structure could simplify the approach)
 - If your teams believe so, state why.
 - If your team believes not, state why not.

Reviewer Comments:

- My team commented that my code may not be the most run-time efficient due to unnecessary loops and recursion that may take longer than using other data structures. There might be a better, more efficient way to write out the code that I thought out in my head. If I had more time, I told my team, I probably would have thought out the logic a bit better and reduced the number of lines needed to write my method. However, one member of my team believed that another data structure could be used to help check the nodes, while another thought that editing the BSTNode, similar to my approach, was the best for run-time efficiency.

Part 3 – Memory – 125 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

The team's assessment of the space efficiency of your proposed solution.

- Does this program use the best Data Structure for memory efficiency given the problem constraints?
 - If your teams believe so, state why.
 - If your team believes not, state why not.
- Does the program make effective use of the heap (i.e., not unnecessarily storing redundant copies of information to produce the solution when a better approach exists), pass/call by reference, function, and method calls?
 - If your teams believes so, state why.
 - If your team believes not, state why not.

Reviewer Comments:

- My team believes that my code takes proper advantage of memory efficiency given the problem constraints. There is no unnecessary elements created besides one variable. Since I only use BSTNode and edit the elements of the class, there is no unnecessary space used besides the space used by searching through the node to determine validity. One of my team thought maybe using an unordered set to help with validating the BSTree, and we think this might not be as space efficient, although it might increase run-time.

Part 4 – Finding Better Solutions – Optional

Let's say that you have reviewed your group mates code and determine there is a better approach than the one you are taking. A concern you might be having right now would be "if I change my answer, is that Academic Dishonesty?" Here is how you can modify your solution post Code Review and maintain academic integrity:

Describe the changes - State the change in "Correctness and Defects", "Performance", and/or "Memory", as appropriate, that contributed to your solution, and how you wrote your own version of the changed code to meet these changes.

Cite: Name your team member or classmate who gave you the idea, to give appropriate attribution.

Coder Comments:

Correctness and Defects

- My team member Connor Ruff gave me the idea of adding an element to the BSTNode class called traverse, which determined in a node had been searched through already. This helped me with Test 2 and Test 5, which deal with whether or not a node had already been searched through, determining a non-valid BSTree.