**Take-Home: Quiz 3 (15 pts) – More with C and Some C++**

Using Canvas <https://canvas.wsu.edu/>, please submit your solution to the correct quiz folder. Your solution should be a .pdf file with the name <your last name>\_quiz3.pdf and uploaded. To upload your solution, please navigate to your correct Canvas ***lab*** course space. Select the “Assignments” link in the main left menu bar. Navigate to the correct quiz submission folder. Click the “Start Assignment” button. Click the “Upload File” button. Choose the appropriate .pdf file with your solution. Finally, click the “Submit Assignment” button.

**Short Answer:**

1. **(9 pts – 1 pt/question)** Given the following fragment of C code, answer the provided questions.

Line 1: int n1 = 10, n2 = 42, list[] = {6, 8, 42, 3, 2, 2, -6};

Line 2: int \* const p1 = &n1;

Line 3: const int \* p2 = &n1;

Line 4: int \* p3 = list;

Line 5: const int \* const p4 = NULL;

Line 6: \*p1 = 15;

Line 7: p1 = &n2;

Line 8: p2 = &n2;

Line 9: \*p2 = 67;

Line 10: p3[4] = 67;

Line 11: list = &n1;

Line 12: p4 = list;

Line 13: \*p4 = 25;

1. (1 pt) Is there any discernible difference between the declared types for p1 and p2 on lines 2 and 3? Briefly explain.
2. (1 pt) Is the assignment operation on line 6 legal? Briefly explain.
3. (1 pt) Is the assignment operation on line 7 legal? Briefly explain.
4. (1 pt) Is the assignment operation on line 8 legal? Briefly explain.
5. (1 pt) Is the assignment operation on line 9 legal? Briefly explain.
6. (1 pt) Is the assignment operation on line 10 legal? Briefly explain.
7. (1 pt) Is the assignment operation on line 11 legal? Briefly explain.
8. (1 pt) Is the assignment operation on line 12 legal? Briefly explain.
9. (1 pt) Is the assignment operation on line 13 legal? Briefly explain.

**Fill-in-the-blank: (note: each question could require multiple word answers)**

1. **(2 pts)** The ability to define multiple functions with the *same* name, but different number, type, and/or order of parameters is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
2. **(2 pts)** A class *declaration* is analogous to a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for a building.
3. **(2 pts)** The ability to *group* data and operations together inside an object is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.