Name:_____ CM: _____ Grade:_____

This quiz is due after you have finished watching the **PointerSyntax**, **PointersInDebugger**, **PointersAsParameters**, **PointersPracticeAndPitfalls,** and **MoreBoxAndPointers** videos. Feel free to complete the quiz as you watch.

Video: PointerSyntax

1. What is a pointer in C?

2. The special character `&`, when placed before the name of a variable (e.g., `&num`), is used to get the _____ of the variable.

Use the following code snippet to answer questions 4 and 5.

```
double grade = 5.0;
double* pGrade;
double x;
```

3. Write a C statement that assigns the address of grade to the variable `pGrade`.

4. Write a C statement that dereferences `pgrade` and assigns the value to the variable `x`, that is, write a statement that assigns the value pointed to by `pgrade` to `x`.

Video: PointersInDebugger

Checkout the `PointersIntro` project in Eclipse.

5. Uncomment the call to `simplePointers()`, and run the program.

   At what location does the program say `num` is stored? _____

   What value of `pNum` is printed? _____

6. Now run the same code in the debugger. What was the value of `pNum` **before** the line `pNum = &num`?

7. Look at the code example that uses change and `pChange`. It uses **another method** of modifying change rather than just saying `change = 0.62`. Explain what we did.

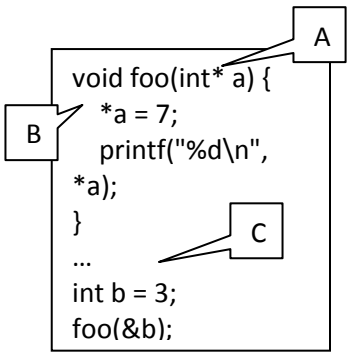
Video: PointersAsParameters

Follow along in the video to fix the broken `downAndp()` function in the `PointersIntro` project.

8. Briefly explain why the function originally didn't work as expected, but did after you fixed it.

For each of the following tasks, indicate the label in the code snippet that accomplishes the task:

- ___ Dereference a pointer to change another variables value
- ___ Declare a parameter to be a pointer
- ___ Pass the address of a local variable to another function



Video: PointersPracticeAndPitfalls

9. Draw the box-and-pointer diagram associated with the following code snippet. Be sure to show intermediate values of variables.

```
int x = 3, y = 5;
int* px = &x;
int* py = &y;
*px = 10;
px = py;
*px = 12;
```

Video: MoreBoxAndPointers

10. Each of the following implementations of swap is wrong. For each, draw a box-and-pointer diagram showing what it does.

- ```
  void swap1(int x, int y) {
      x = y;
      y = x;
  }
  ```

- ```
  void swap2(int x, int y) {
      int temp;
      temp = y;
      y = x;
      x = temp;
  }
  ```

- ```
  void swap3(int* x, int* y) {
      int* temp;
      temp = y;
      y = x;
      x = temp;
  }
  ```

11. After watching this set of videos, what questions, if any, do you have? If none, please write "None".
  ```